SENSORY®

NLP-5x

NATURAL LANGUAGE PROCESSOR WITH MOTOR, SENSOR AND DISPLAY CONTROL

Data Sheet

General Description

The NLP-5x is Sensory's new Natural Language Processor targeting consumer electronics. The NLP-5x is designed to offer advanced speech recognition, text-to-speech (TTS), high quality music and speech synthesis and robotic control to cost-sensitive high volume products. Based on a 16-bit DSP, the NLP-5x integrates digital and analog processing blocks and a wide variety of communication interfaces into a single chip solution minimizing the need for external components.

The NLP-5x operates in tandem with FluentChip™ firmware - an ultra-compact suite of technologies that enables products with up to 750 seconds of compressed speech, natural language interface grammars, TTS synthesis, Truly Hands-Free™ triggers, multiple speaker dependent and independent vocabularies, high quality stereo music, speaker verification (voice password), robotics firmware and all application code built into the NLP-5x as a single chip solution.

The NLP-5x also represents unprecedented flexibility in application hardware designs. Thanks to the highly integrated architecture, the most cost-effective voice user interface (VUI) designs can be built with as few additional parts as a clock crystal, speaker, microphone, and few resistors and capacitors. The same integration provides all the necessary control functions on-chip to enable cost-effective man-machine interfaces (MMI) with sensing technologies, and complex robotic products with motors, displays and interactive intelligence.

The NLP-5x is supported by a full suite of tools including IDE/compiler/debugger, Demo/Emulation Board, Programming/Verification Board(for the OTP), Rapid Prototyping Modules (RPM), acclaimed QuickT2SI for recognition set and grammar creation in minutes, and QuickSynthesis for prompts and music.

Features

Broad Range of FluentChip™ Technologies

- **WORLD'S FIRST NATURAL LANGUAGE INTERFACE ON AN EMBEDDED PROCESSOR!**
- *Real world* noise-robust Speaker Independent (SI) & Dependent (SD) recognition; up to 75 word sets
- TTS (text-to-speech) for text-based speech playback
- Phrase spotting of up to 30 commands or key phrases embedded in speech
- Truly Hands-Free triggers (using phrase spotting) alert the recognizer to listen for commands
- Speaker Verification (SV) – Noise robust voice password biometric security
- High quality stereo MP3 decoder
- 24-voice MIDI-compatible stereo music synthesis; allows simultaneous speech synthesis
- High quality, 2.4-64 kbps speech synthesis & sound effects with Sensory "SX" technology
- SonicNet for acoustic data networking
- Support for major EU and Asian languages for speech recognition and TTS

Integrated Single-Chip Solution

- Powerful 16-bit DSP – up to 80MHz clock; single issue/single MAC; nested interrupts; on-chip debug
- 128KBytes One-Time-Programmable (OTP) memory for code and constants
- Three 16-bit ADC channels - two with pre-amps (standard or beam-forming microphones) & one line input for sensors
- Two 16-bit DAC channels - stereo music, speech and sound effects at 48KHz sample rate
- Mono PWM direct speaker drive
- 24 KBytes total RAM, including on chip storage for SD and SV templates
- Five timers (4 GP, 1 Watchdog), plus Real Time Clock and on-chip 128KHz RC oscillator scalable to 32KHz
- LCD control logic and drive - up to 104 icons or pixels; SPI for large array driver interfaces
- Motor control logic – up to 3 bi-directional motors
- USB1.1, SPI, UART-Lite and infrared (IR) interface
- Analog comparator unit (4 inputs)
- Power-on Reset and Brown-out Reset
- Low EMI design for FCC and CE requirements
- 40 configurable general purpose I/O lines; 10 mA or 22mA (typical) output drive and 3.6V tolerance
- External 16 bit data and 23 bit address bus with 3.6V tolerance; configurable as 39 more GPIO

Long Battery Life

- 1.8V +/- 10% operation; includes on-chip regulation of 2-3.6V Vdd input
- 16-36mA typical operating current at 1.8V

- 50µA typical sleep current

Table of Contents

General Description 1
Features 1
Table of Contents 2
NLP-5x Overview 4
FluentChip™ Technologies 5
NLP-5x Architecture 6
Reference Schematic 10
..... 10
NLP-5x Processor 11
NLP-5x Memory Architecture 11
Processor Internal Control Registers 13
Instruction Set 16
NLP-5x Memory 17
Physical Memories 17
Instruction Space 18
Data Space 22
External Memory Interface 25
NLP-5x Reset, Oscillators, Clocks, Power Management 30
Reset 30
Oscillators 31
Clocks 35
Turbo Mode 37
System Power Control and Wake Up 37
Sleep Mode 38
I/O Wake Event 41
Miscellaneous Interrupts, System Control 42
Host/Peripheral Interface Port 45
Watch Dog Timer 46
Interrupts 47
Merged Interrupts 51
Interrupt Vector Table 53
General Purpose I/O and Timers 57
Ports 0, 1, 2 57
Auxiliary Ports XP0, XP1, XP2 58
Alternate I/O Functions 60
General Purpose I/O Equivalent Circuits 62
General Purpose Timers 69
Audio and Analog Circuitry 74

Analog Front End....74
Analog Front End Block Diagram....75
Programmable Gain Amplifier....76
Line-level Channel....78
A/D Converter (ADC)....79
Pulse Width Modulator (PWM)....87
Comparator Unit....91
Stereo D/A Converter (DAC)....94
Serial Communications....97
UART....97
Synchronous Serial Port (SSP)....101
Infrared Interface....110
USB 1.1 Interface....116
LCD Interface....120
Motor Controllers....126
PWM Section....126
Sensor Section....129
APPENDIX A - Instruction Set Summary....132
APPENDIX B - Peripheral Registers Summary....139
APPENDIX C – Package, Pin, and Pad Definitions....144
LQFP 176 Package Pin Assignments....144
Pin Functions....145
Die Pad Ring....150
NLP-5x Die Bonding Pad Locations....151
Mechanical Data....152
APPENDIX D – Preliminary Electrical Characteristics....154
....161
The Interactive Speech™ Product Line....162

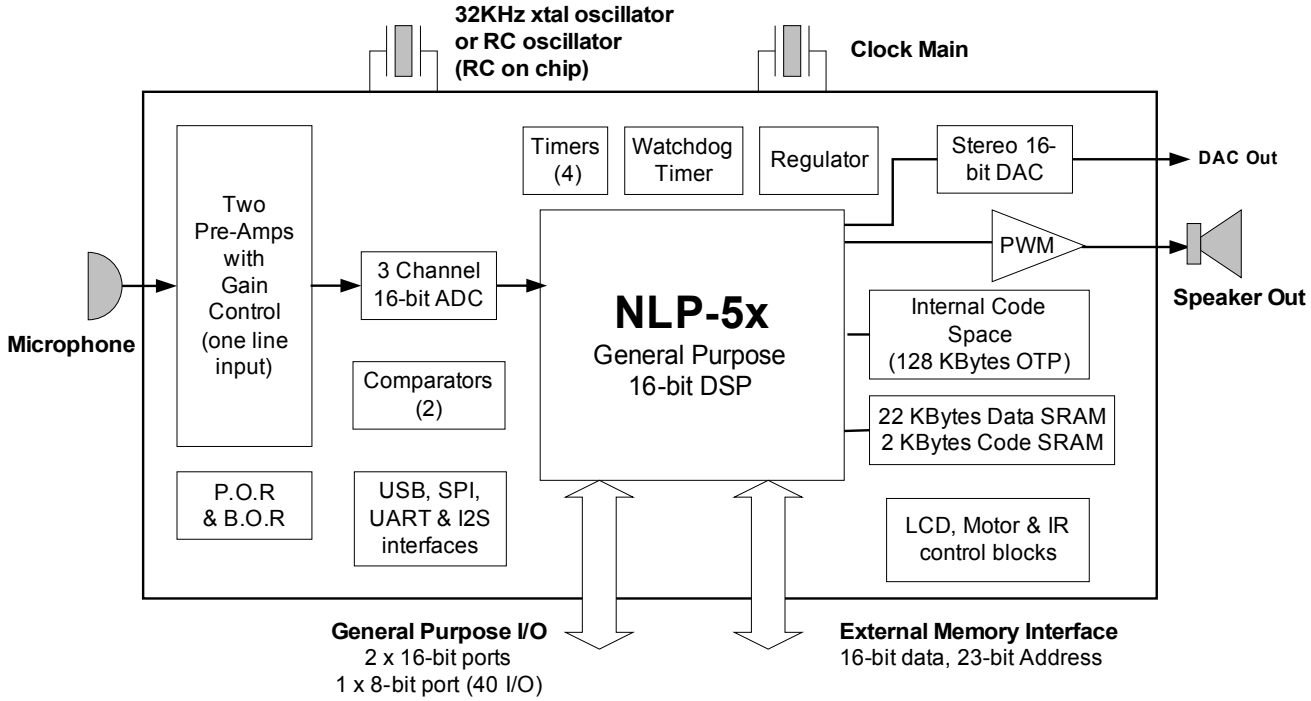
NLP-5x Overview

The NLP-5x is the newest member of the Interactive Speech™ line of products from Sensory. It features a high-performance 80MHz 16-bit DSP with on-chip ADC, hi-fidelity stereo DAC, microphone preamplifiers, RAM, OTP code and constant memory, and many kinds of peripheral interfaces and control blocks. The NLP-5x is designed to offer a high degree of integration and versatility to low-cost, power-sensitive applications in appliances, robotics, telephony and automotive hands-free markets. Many capabilities have been integrated in the NLP-5x to reduce total system cost and increase system reliability.

The NLP-5x operates in tandem with FluentChip™ firmware, an ultra compact suite of recognition and synthesis technologies. This reduced software footprint enables, for example, products with 3 minutes of compressed speech, multiple speech recognition vocabularies, voice password, and all application code built into the NLP-5x as a single chip solution. Revolutionary flexible grammars allow the user to say multiple commands in a single phrase, and even in a flexible order. This results in the most natural use of speech recognition! Many more technology offerings in FluentChip™ support development of entertaining robotics features for animation, display, networking, music, etc.

Along with the vast array of on-chip features, the NLP-5x provides a very cost-effective mixed signal platform for general-purpose applications and development of custom algorithms. The full suite of industry standard tools for easy product development makes the NLP-5x an ideal platform for consumer electronics.

NLP-5x Block Diagram



FluentChip™ Technologies

The **NLP-5x** is designed to support Hidden Markov Modeling (HMM) as well as Neural Network (NN) technologies provided in FluentChip™ firmware to perform speaker independent (SI) speech recognition. A brief summary of **Speech Recognition** technologies include:

- **Natural Language Interface** provides the unique ability to understand context-specific user's commands in the natural way the user would like to speak. Order independence allows flexibility in commands and speech prompts can request any missing information (form filling).
- **Speaker Independent** recognition requires no user training. The NLP-5x can recognize up to 75 words in an active set (number of sets is limited only by internal ROM or external memory size). Text-to-SI (T2SI) recognition, based on HMM technology, allows creation of SI recognition sets in seconds.
- **Speaker Dependent** recognition allows the user to create names for products or customize vocabularies. Up to 50 words can be recognized in an active set (number of sets is limited only by internal ROM or external memory size). The NLP-5x can store up to 10 SD words in on-chip SRAM.
- **Truly Hands-Free** phrase spotting allows the chip to continuously listen for triggers or commands, even in the presence of high noise. In phrase spotting mode, the word(s) to be recognized may be spoken in the middle of speech, for Truly Hands-Free™ operation.
- **Speaker Verification** allows the NLP-5x to identify whether a pre-determined and trained word or phrase is spoken by the original speaker, enabling applications that require voice passwords. Up to 10 SV templates can be stored on-chip, or more with external programmable memory.

Speech/Music Synthesis & Audio Functions

The NLP-5x processing power, high-quality 16-bit ADC and stereo DAC support a variety of audio and speech playback technologies:

- Text-to-speech (TTS) is supported for systems requiring text-based speech playback, and requires less than 1MByte of external memory. TTS works well for names or text/phrase reading and is supported in multiple languages.
- Speech compression and playback using state-of-the-art "SX" technology. Data rates vary from 2.4K to 64K bits per second depending on desired quality, frequency response and available memory storage. SX uses on-chip or off-chip ROM to store speech data
- Stereo MP3 with a 5-band equalizer.
- MIDI synthesis for high-quality, 24-voice, stereo wave-table music.
- Record & Playback at various compression levels for voice memo functions
- Pitch Detect, which enables talking and singing in character voices, etc
- LipSync automates animation of a motorized mouth in sync with speech
- BeatPredict automates animation of movement, etc in time with music beat

Sensors, Touch and Gesture Technologies with Sensory Analysis

- On chip firmware for HMM's can be used for complex signal analysis and pattern recognition from sensors detecting visible light, IR, electric fields, mechanical changes, capacitive and resistive changes, accelerometers and more.
- DSP processing, mixed signal I/O and a broad variety of communication protocols support interfacing with external sensors and various data conversion and interpretation.

Sensory and 3rd party developers provide support for presence detection, touch and position sensors, gesture and motion analysis, etc. Combined with voice user interface capabilities, this enables man-machine interface solutions with an unprecedented combination of power and cost-effectiveness.

- SoundSourcing – with 2-3 microphones a product can locate a speaker and turn or move towards the person talking.
- IR Presence Detect – detects up to 6 feet, triggering speech playback, speech recognition, etc.

Telecommunications

The NLP-5x is capable of implementing voice codecs suitable for cell phone or PDA mobile communications. It can implement the companding function along with speech recognition, acoustic echo cancellation, DTMF encode/decode, VOIP protocols and on-chip voice compression, easing the addition of speech command and control, and answering machine functions to telephony applications. The NLP-5x is also capable of implementing a beam forming microphone array.

NLP-5x Architecture

Please refer to the block diagram of the NLP-5x on page 9 while reading this section.

The NLP-5x has a 16-bit DSP processor core running up to 80MHz. The core includes a set of sixteen (16) 16-bit *operand registers*, used as the source of instruction operands and as the destination for results of operations, and a set of twenty-six (26) 16-bit *control registers*, used to configure the processor. Additional memory-mapped *peripheral registers* external to the core are used for peripheral and system support. (For a summary, see Peripheral Registers Summary, pg 129).

The NLP-5x has three independent memory busses: the high-speed 32-bit-wide instruction bus, the high-speed 32-bit-wide data bus, and the 16-bit-wide expansion bus. There are five independent physical memories on-chip plus an external memory bus: instruction OTP, instruction RAM, data RAM, expansion OTP, and a small RAM shared with the USB controller.

The instruction OTP is 32K words, organized as 16K x 32 bits, and is connected to the high speed instruction bus (note: "word" will mean 16 bits when used in this document). The instruction OTP is 32 bits wide because of the ability of the processor to fetch two 16-bit instructions in one cycle. The instruction OTP can be read as fast as 40 MHz. When the NLP-5x is configured to run faster than 40 MHz (see Turbo Mode, pg. 36), the processor will have to add a wait state when reading from the instruction OTP. This means execution from the instruction OTP will not be as fast as from the instruction RAM for clock rates above 40 MHz.

The instruction RAM is 1K words (2K bytes), organized as 512 x 32 bits. Instructions in this RAM can run at the full speed of the core, that is, up to 80 MHz. This RAM is connected to the high speed instruction bus.

The data RAM is 11K words (22K bytes), organized as 5.5K x 32 bits, and is connected to the high speed data bus. This memory is 32 bits wide because of the ability of the processor to read or write two adjacent words simultaneously. Programs that read and write to this data RAM can run at the full speed, that is, up to 80 MHz.

From the programmer's perspective, the fact that the instruction and data busses are 32 bits wide is not visible. But the programmer should be aware that programs using memories connected to these busses will run faster than those busses using expansion memory, in part because they are twice as wide.

The expansion OTP is 32K words, organized as 32K x 16 bits, which is the width of the NLP-5x expansion data bus. The internal expansion OTP may contain a mix of executable instructions and constant data. Combined with the instruction OTP, the total on-chip OTP is 64K words (128K bytes).

The expansion OTP and the instruction OTP can be disabled by connecting the pin -XM low. This allows code to be executed from external memory, but with substantially reduced speed. The instruction RAM is still available when –XM is low.

There is a 256-word RAM shared with the USB controller. When this RAM is not needed by the USB controller it is available for general use.

The NLP-5x has an Expansion Memory Controller that controls the external memory interface for reading and writing to off-chip RAM, flash memory, or other devices with a parallel interface. The external address bus has 23 address bits, 16 data bits, plus two chip selects, allowing external addressing of two 8MW external memory devices. There are provisions for programmable wait states if required. (see: External Memory Interface, pg. 25).

The Expansion Memory Controller also handles the instruction bank switching and data segmentation used to map instruction and data address ranges ("windows") to either the on-chip expansion OTP or external memories. It also provides access to NLP-5x peripheral registers.

The on-chip OTP memories can be programmed either with a parallel interface or via a JTAG port. The JTAG port can also be used for in-circuit debugging (see *Programming the NLP-5x Design Note (80-0327-1*).

The NLP-5x has thirty-two general purpose I/O pins (port 0 and port 1) that can be individually programmed to be inputs (with or without pull-up devices) or outputs. Many of these pins have alternate functions. There are also eight more I/O pins (port 2) that can be used for either serial communications or general purpose I/O, bringing the total GPIO pin count to forty. In addition, some or all of the address and data pins of the external memory interface can be configured for general purpose I/O (see: General Purpose I/O, pg. 54).

Some of the GPIO can be configured to drive an LCD display with up to 104 pixels, segments or icons, with up to 4 commons.

Some of the GPIO can be configured for motor control, with as many as 3 bi-directional, pulse-width-modulator (PWM) units and 3 interrupter- based tachometer sensor units, enabling full control of up to 3 motors.

The NLP-5x has a high-speed crystal-controllable oscillator, OSC1, and two low-speed oscillators: OSC2 is a crystal-controlled oscillator. OSC3 is an on-chip RC oscillator.

OSC1 is designed to operate at frequencies from 2.0 MHz to 8.0 MHz. The recommended frequency is 4.000 MHz. The OSC1 oscillator can be connected to a crystal or ceramic resonator, or it can be driven from an external clock source.

A number of other clocks are derived from OSC1 via an on-chip programmable PLL. The PLL multiplier is programmable to provide a system clock (PLLCLK) up to 80MHz. The PLL can also be bypassed with no multiplication, in which case PLLCLK is the same rate as the OSC1 rate. The bypass state is the default state after reset. A second output of the PLL is the 48MHz clock for the USB controller block.

OSC2 operates at 32.768 KHz. The OSC2 oscillator may be controlled by a crystal for accurate timekeeping applications.

The OSC3 oscillator operates at 128 KHz and should achieve +/- 5% frequency tolerance *after software adjustment*. Six bits of software trim are provided to achieve the RC tolerance. There is an option for the OSC3 oscillator to be divided by 4 to produce 32 KHz +/- 5%.

As in previous Sensory chips, the NLP-5x allows either of the two low-speed oscillators to be the source for the processor clock, CPUCLK. The OSC2 or OSC3 oscillators can also used for the Watchdog Timer circuit, the LCD controller, or Timer #2.

The NLP-5x processor has two built-in general purpose 16-bit timers, Timer #0 and Timer #1. The NLP-5x implements two other additional general purpose 16-bit timers. Timer #2 is driven by one of the two low frequency oscillators. Timer #3 is driven by CPUCLK.

The NLP-5x I/O pins can operate over a wide range of voltages - from 1.62V to 3.60V. However, the digital core of the NLP-5x requires 1.8V +/- 10%. The power supply for the core can be provided by the integrated, low-headroom voltage regulator, or by an external 1.8V power supply.

The NLP-5x can be placed into a low power “sleep” state. (see: System Power Control and Wake Up, pg. 36) Software can determine which oscillators are stopped while in sleep mode. During sleep, the processor will not be running. Software determines which events can wake the chip from sleep mode. Some events require OSC1, OSC2 or OSC3 to be enabled during sleep. The wake-up time depends on whether the oscillator used by the processor is halted during sleep.

To reduce current even further during sleep mode, there is an option to disconnect the power supplies to the data RAM and one half (odd addresses) of the instruction RAM. In this case the contents of these RAMs will be lost. Power to the USB RAM and half of the instruction RAM will not be disconnected so they can be used to hold state data. There is a further option to put the on-chip regulator in a low power state. In this state the current output capability of the on-chip regulator is reduced, and the tolerance is relaxed.

The NLP-5x has an internal A/D converter (ADC) with 16 bits of resolution that can be configured with one, two, or three input channels. Time multiplexing is used when the ADC is configured for more than one input channel.

The maximum ADC rate is 96K samples-per-second for each of 3 channels, for a combined conversion rate of 288K samples-per-second (see: A/D Converter, pg. 69).

The NLP-5x has two microphone preamplifier circuits, each with 16 levels of programmable gain in each of 2 ranges. Each has a zero crossing detector. The preamplifiers amplify signals from electret microphones to the level required by the ADC (see: Programmable Gain Amplifiers, pg.66).
The NLP-5x has a stereo 16-bit D/A converter (DAC) for high quality sound production. Each DAC channel has a pair of digital, differential outputs from a delta-sigma modulator. The differential signals require off-chip filtering and amplification (see Stereo D/A Converter, pg. 84).

The NLP-5x has one audio Pulse Width Modulator (PWM) channel for sound output that can directly drive an 8-ohm speaker (see Pulse Width Modulator, pg. 77).

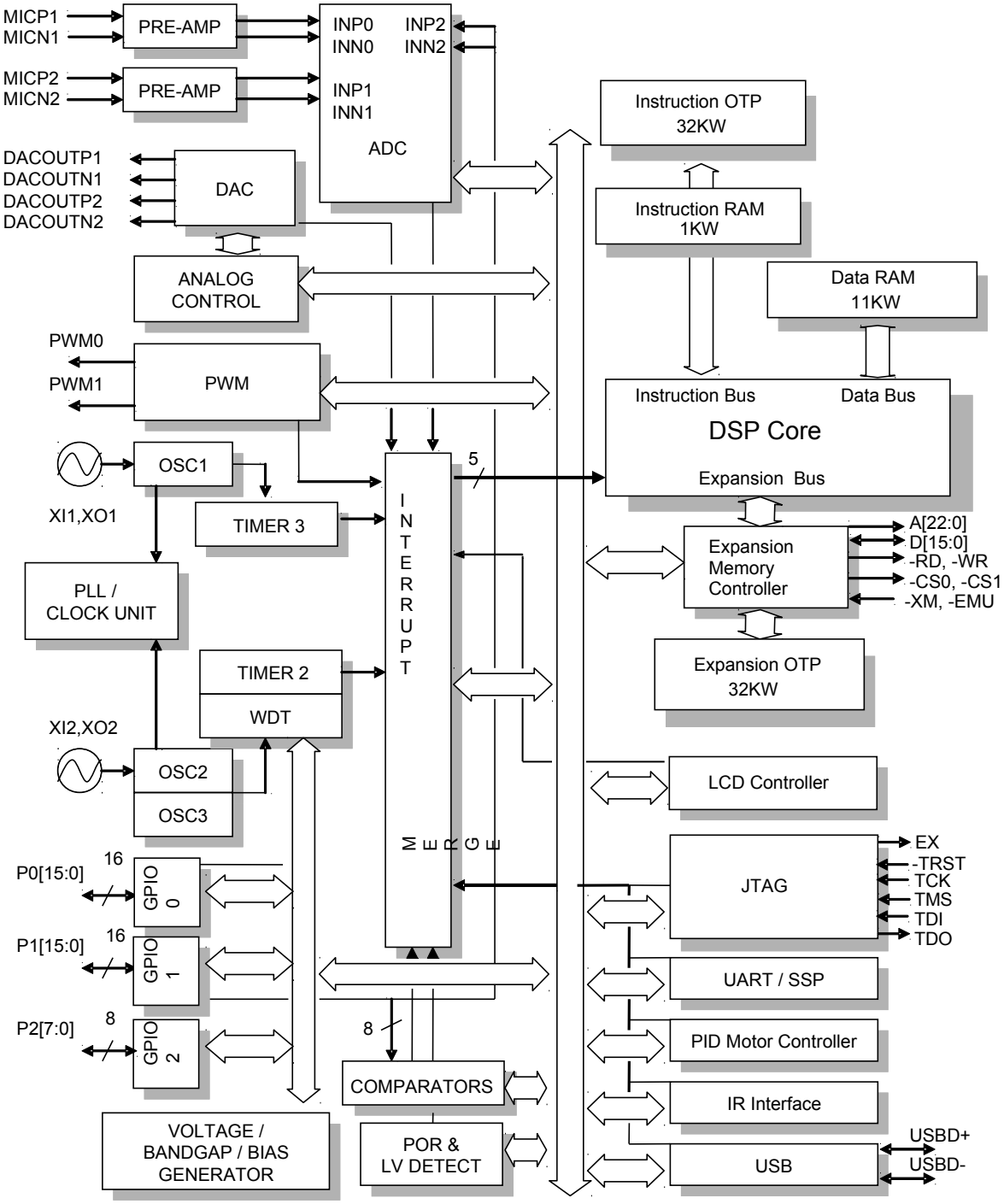
There are several serial ports (see Serial Communications, pg 87). There is a UART and a synchronous serial port with several modes, including I^2S slave, SPI master, and SPI slave. The NLP-5x has a USB 1.1 controller with two endpoints and an integrated USB physical interface. There is support for infra-red serial transmitter and receiver.

The NLP-5x -EMU pin can put the chip into a special emulation mode to create a RAM-based NLP-5x emulator to assist in software development. In emulation mode the NLP-5x becomes a “slave” bus-connected device. The NLP-5x’s processor is powered down, but the peripherals and analog functions are active. A standard chip version of the DSP core is used to emulate the NLP-5x processor. (see *Programming the NLP-5x Design Note #80-0127-1*).

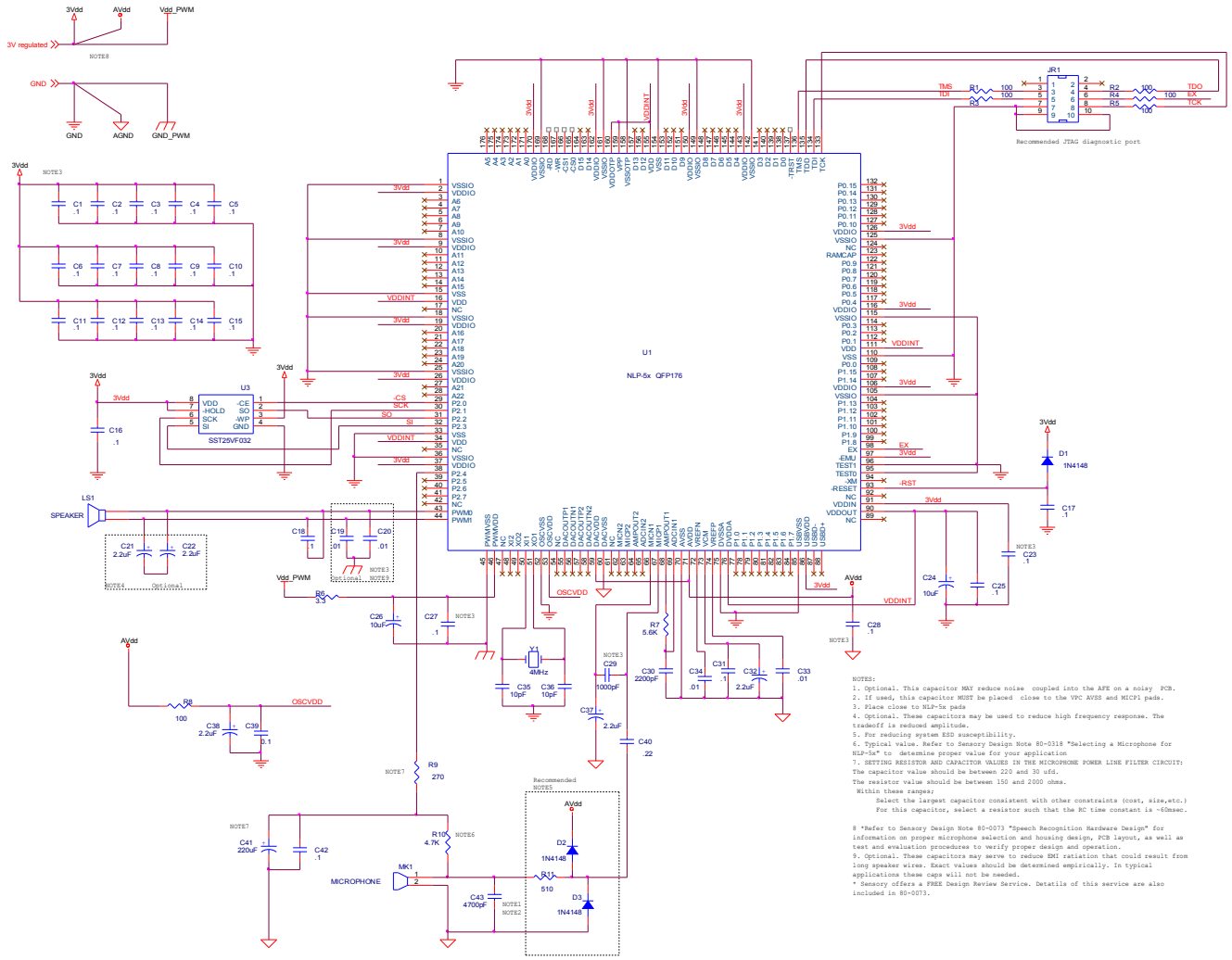
This datasheet documents specifications for 0C to 70C. However, the NLP-5x is designed to work over the range of –40C to +85C. Please contact Sensory Sales for pricing and specifications of this expanded temperature range.

The NLP-5x will be sold as a tested die, and also packaged in a 176-pin LQFP package. (see: Pad, Package, and Pin definition, pg 134).

NLP-5x Block Diagram



Reference Schematic



NLP-5x Processor

The NLP-5x processor is based on a 16-bit DSP core. This core is a Harvard-architecture, RISC processor with a load/store instruction set.

NLP-5x Memory Architecture

The NLP-5x's Harvard architecture means that there are separate memory spaces for instructions and data. To suit this architecture the NLP-5x has separate high-speed instruction and data busses. The NLP-5x does provide a mechanism for routing data space reads and write operations to instruction memories on a temporary basis, such as when loading opcodes into instruction RAM.

Beyond the above-mentioned instruction and data busses, the NLP-5x has a third bus, the Expansion Bus, which can be adapted to memories of any speed, on- or off-chip. The NLP-5x core is wired to know which address ranges of instruction space and data space are to be directed to the Expansion Bus. As implemented in the NLP-5x, the Expansion Bus is shared by both instruction space and data space accesses, so almost any memory connected to the Expansion Bus can be accessed either as code or data.

The NLP-5x has 16-bit words for instruction opcodes and data items. All memories are addressable with a resolution of 16-bit words. Some busses are two words wide to read or write two words simultaneously.

The NLP-5x has a 16-bit memory address range for both instruction space and data space. Expansion of either memory space beyond 64K words requires addition of banking mechanisms. The NLP-5x has separate banking mechanisms for instruction space and data space.

In the NLP-5x, the DSP core knows about 5 different memory categories, each defined by location and method of access.

Memory Category	Description	Method of Access
Operand Registers	16 general purpose 16-bit registers, r0 to r15	Direct operands to, and results of, arithmetic, logical, transfer, and other processor operations.
Control Registers	26 16-bit registers with control functions. Control registers have names beginning with '%'.	Accessed via transfer to or from an operand register, or in some cases, from immediate data encoded in the opcode, or via bit set, bit clear, bit invert, and bit test instructions. Some instructions transfer one control register to another.
Instruction Bus Memories	Memories connected to the 32-bit-wide bus that can access two instructions in a single read cycle.	Accessed via instruction fetches for addresses starting at 0 but *below* the instruction bus limit (8000h if -XM= 1, 1000h if -XM= 0). Can also be accessed as data space if **lis** or **sis** bits in %smode are set.
Data Bus Memories	A 32-bit wide data bus that can access two data words in a single read or write cycle.	Accessed via data space load and store operations for those addresses starting at 0 but *below* the data bus limit (4000h).
Expansion Bus Memories	A 16-bit-wide data bus.	Accessed via instruction fetches for addresses *at or above* the instruction bus limit (8000h if -XM=1, 1000h if -XM = 0), or via data loads and stores directed to addresses *at or above* the data bus limit (4000h).

Instruction Bus Memories and Turbo Mode
In the NLP-5x, the instruction bus has both RAM (the Instruction RAM) and OTP (the Instruction OTP) memories. When accessing the Instruction RAM, the instruction bus can run up to the full rated speed of the processor clock, CPUCLK, namely 80 MHz. The Instruction OTP, in contrast, has a maximum access rate of 40 MHz. If CPUCLK is selected to be above 40 MHz, a single wait state is required when the core reads from Instruction OTP, and thus the chip must be in Turbo Mode (pg 36). In this mode a single wait state is inserted for fetches from Instruction OTP. Turbo mode is selected via bits [2:0] of register clkSelect.

Data Bus Memories
In the NLP-5x, the only memory connected to the data bus is the data RAM, which can operate up to the maximum CPUCLK rate of 80 MHz.

Expansion Memories
The DSP core inside the NLP-5x has two sets of hard-wired configuration inputs that define address limits for the instruction bus and data bus. Accesses below these address limits are directed to either the high speed instruction bus or the high speed data bus. Accesses at or above the address limits are directed to the Expansion Bus. The Expansion Bus has a handshake mechanism to allow hardware implementation of any number of wait states for reads and writes. The NLP-5x has implemented its Expansion Memory Controller to process expansion bus read and write requests and to direct each request to on-chip memories (the Expansion OTP, USB RAM, or Peripheral Registers), or off-chip memories via the external memory bus. Bits in the cBank, dSeg0, and dSeg1 registers are used to determine whether expansion memory requests are directed to on-chip or off-chip memories.

The NLP-5x's implementation of the memory architecture is described in more detail in the chapter NLP-5x Memory (pg 17).

Instruction Cache
The DSP core inside the NLP-5x has an 8-word instruction cache. Execution of code from the cache can run at full speed. Thus small loops can run very efficiently, even if the source of the instructions was from external, slower memories

Stack
The NLP-5x provides no hardware stack. Stack management is done in software. The C compiler uses register **r12** as the stack pointer, with the stack occupying the uppermost part of the Data RAM.

Processor Internal Control Registers

The NLP-5x core has 26 *control registers*, which control the state and operation of the processor core. These are distinct from the 16 *operand registers*, r0-r15. Following are selective descriptions of some specific control registers that might be encountered by programmers who write mostly in C.

Control Register Name	Description	Further Details
%amode	Address Mode Register	pg. 13
%fmode	Functional Mode Register	pg. 13
%hwflag	Hardware Flag Register	pg. 14
%imask	Interrupt Mask Register	pg. 55
%ip0, %ip1	Interrupt Priority Registers	pg. 56
%ireq	Interrupt Request Register	pg. 55
%smode	System Mode Register	pg. 15
%tc	Timer Control Register	pg. 70
%timer0, %timer1	Timer 0 and 1 Counter Reload Registers	pg. 71

Address Mode Register (%amode)
This register is used to enable bit-reversed addressing modes which can be useful for certain algorithms, such as the Fast Fourier Transform (FFT). C programmers will normally not encounter this control register. Interrupt service routines typically make no assumption about the state of this register, so among the first things they do is save this register on the stack and clear it, so loads and stores behave in the normal manner.

Functional Mode Register (%fmode)
The %fmode register contains the functional mode bits for the arithmetic and logic unit (ALU) of the NLP-5x. For the most part, C programs assume this register is zero. Interrupt service routines typically make no assumption about the state of this register, so among the first things they do is save this register on the stack and clear it (or set it to the desired functional mode) so that arithmetic and logical instructions behave as expected. An interrupt service routine that does not do any arithmetic does not need to modify this register.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
reserved										rez	sat	res	q15	sre	mre

%fmode is reset to zero.

Bits [15:6] Reserved. These bits should be written with zeros.

Bit [5] Round Zero Enable. If bit 0 of %fmode is zero, this bit has no effect. If this bit is set and bit 0 of %fmode is also set, then the lower 16 bits of 32-bit results is cleared to zero. Affected instructions: mac, macn, mul, muln, mac2, cmacr, cmaci, cmulr, cmuli, dmac, dmul, and round.e.

Bit [4] If set, the result of affected arithmetic operations saturate to maximum positive or negative values.

Bit [3] Reserved.

Bit [2] Q15 mode. When set, the product of MAC and MUL instructions are shifted left one position. The special case of -32768 x -32768 in Q15 mode produces the maximum positive value.

Bit [1] Shift Round Enable. When set, if the last bit shifted out by a SHRA(.E) instruction is a 1, the result is incremented by 1.

Bit [0] MAC/MUL Round Enable. For 16-bit MUL instructions, 0x8000 is added to the result. For 16-bit MAC instructions, 0x8000 is added to the product before summing to the target accumulator. For 32-bit DMUL instructions, 0x8000.0000 is added to the 64-bit result prior to shifting the result right 32 places.

Hardware Flags Register (%hwflag)

The %hwflag register contains condition flags that occur as the result of various instructions or processor status. The sticky overflow flags, sv and gsv, can only be cleared through software by writing zeros to these bits.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
reserved					v	gv	sv	gsv	c	ge	gt	z	ir	ex	er

Undefined after reset.

Bit [10] 32-bit Overflow. This bit can be set by MAC, ADD and SHLA instructions if the result exceeds the range of 32-bit signed numbers. Cleared by MUL in all cases.

Bit [9] Guard Register (40-bit) Overflow. Similar to bit 10, but only occurs on MAC instructions when the 40-bit combination of %guard register and the 32-bit accumulator exceeds the range of 40-bit signed numbers.

Bit [8] Sticky Overflow. Set when bit 10 is set, cleared only by software.

Bit [7] Sticky Guard Register Overflow. Set when bit 9 is set, cleared only by software.

Bit [6] Carry flag. Not set by shifts. Cleared by MUL. For subtract and compare, the carry flag is set as follows.

For the SUB instruction (or CMP) below:

```
sub  reg_dest, reg_src
```

Calculates reg_dest – reg_src. Carry flag is set based on the unsigned comparison when:

```
reg_dest >= reg_src
```

and cleared otherwise.

Thus branch on carry (BC) is the same as *branch on unsigned greater than or equal*, and branch on no carry (BNC) is the same as *branch on unsigned less than.*

Bit [5] Greater than or equal. Set if the result of the operation is positive.

Bit [4] Greater Than. Set based on signed comparisons. Not affected by MUL.

Bit [3] Zero Flag. Not affected by MUL.

Bit [2] Interrupt Pending Flag. Set if any interrupt is pending, regardless of the mask or priority level of the interrupt.

Bit [1] Emulation Transmit Flag – this bit and bit 0 are used for handshake signals for debug communication via the JTAG port. The state of this bit is reflected by the state of the output pin EX.

Bit [0] Emulation Receive Flag – this bit and bit 1 are used for handshake signals for debug communication via the JTAG port.

System Mode Register (%smode)

The %smode register controls the DSP core system modes. Not all of these bits are used in typical NLP-5x applications.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
lvl2	lvl1	lvl0	res	dct	fie	ict	dsb	uvt	us	lis	sis	cb0	cb1	dir	ddr

%smode is reset to zero.

Bits [15:13] The lvl[2:0] bits control the clock inputs of the NLP-5x core. The only valid combination of these bits supported by the NLP-5x is 000, the normal operational mode. Any other combination may result in unpredictable behavior. Power and clock management in the NLP-5x is done via peripheral registers rather than via lvl[2:0].

Bit [12] Reserved. This bit should always be written with zero.

Bit [11] Data Cache Invalidate. Inverting this bit invalidates the contents of the data cache. The value of the dct bit does not indicate valid or invalid cache contents. In normal situations this bit should always be written with zero.

Bit [10] This bit is not supported by the NLP-5x and should by written with zero.

Bit [9] Instruction Cache Invalidate. Inverting this bit invalidates the contents of the instruction cache. The value of the ict bit does not indicate valid or invalid cache contents. In normal situations this bit should always be written with zero.

Bit [8] Not used by the NLP-5x. This bit should be written with zero.

Bit [7] This bit selects the location of the interrupt vector table:
1 Interrupt vector table starts at 0x0000
0 Interrupt vector table starts at 0xF800
The NLP-5x technology software expects the interrupt vector table to be at 0xF800.

Bit [6] Uniscalar Mode.
Not used by the NLP-5x. This bit should be written with zero by programs running on the NLP-5x.
***For the emulator using the VSI403LP processor, this bit should be set high*.**

Bit [5] lis – Load from Instruction Space:
1: data space reads are directed to instruction space.
0: normal operation

This bit can be used to read data from memories connected to the instruction bus. If interrupt service routines make the assumption that this bit is clear, interrupts should be disabled whenever this bit is set.

Bit [4] sis – Store to Instruction Space:
1: data space writes are directed to instruction space.
0: normal operation

This bit can be used to write data to the instruction RAM memory connected to the instruction bus. If interrupt service routines make the assumption that this bit is clear, interrupts should be disabled whenever this bit is set.

Bit [3] Circular Buffer 0 Enable. If interrupt service routines make the assumption that this bit is clear, interrupts should be disabled whenever this bit is set.

Bit [2] Circular Buffer 1 Enable. If interrupt service routines make the assumption that this bit is clear, interrupts should be disabled whenever this bit is set.

Bit [1] dir – disable Instruction RAM/Instruction OTP. If set, instruction fetches (or reads/writes with lis/sis bit set) that normally would be directed to the Instruction Bus are directed to the Expansion Bus instead. If interrupt service routines make the assumption that this bit is clear, interrupts should be disabled whenever this bit is set.

Bit [0] ddr – disable Data RAM. If set, data reads and writes that normally would be directed to the Data Bus are directed to the Expansion Bus instead. If interrupt service routines make the assumption that this bit is clear, interrupts should be disabled whenever this bit is set.

Instruction Set

The instruction set of the DSP core is summarized in this document in Appendix A, pg. 122.

NLP-5x Memory

Physical Memories

The NLP-5x has five independent physical memories on-chip: Instruction RAM, Instruction OTP, Data RAM, Expansion OTP, and a small RAM shared with the USB controller. The NLP-5x also provides access to off-chip parallel and serial memories. In the first version of the NLP-5x, the OTP memories are implemented as EPROMs. Unless packaged as windowed parts, these EPROMS are effectively OTP memories (one-time-programmable). In windowed packages, the EPROMs can be erased with UV light.

All addresses are "word addresses", that is, they are intended to read or write 16-bits at a time. Byte addresses are not supported by the NLP-5x.

The Instruction RAM is 1K words arranged as 512 x 32 bits. It is connected to the Instruction Bus memory interface of the core. This RAM is accessible regardless of the state of the -XM pin (i.e., even when -XM is pulled low to disable the internal Instruction OTP, the Instruction RAM is still enabled). The Instruction RAM appears at addresses 0000h to 03FFh of instruction space. It can also be read and/or written in the same manner as data space memory by setting the **lis** and/or **sis** bits of the %smode register, which cause load and/or store operations to be routed to instruction space.

The Instruction OTP is 32K words arranged as 16K x 32 bits. It is connected to the Instruction Bus memory interface of the core. When the -XM pin is high (or no-connect), the uppermost 31KW of this memory will appear in logical addresses 0400H-7FFFH of instruction space. The first 1KW (0000h to 03FFh) is normally not accessible because that address range is assigned to the Instruction RAM. Bit 11 of peripheral register miscCtrl can be set to disable the Instruction RAM for reading (it still is enabled for writing), and make this first 1KW of Instruction OTP accessible.

The Data RAM is 11K words arranged as 5.5K x 32 bits. It is the only memory connected to the core's Data Bus memory interface. The Data RAM will appear in data space addresses 0000h to 2BFFh. Future versions of the NLP-5x may expand Data RAM up to 16KW (0000h to 3FFFh). Accesses to Data space addresses at or above address 4000h are directed to the Expansion Bus. The behavior of the NLP-5x in response to accesses to unpopulated addresses 2C00h to 3FFFh is undefined.

The fourth on-chip memory is the 32KW Expansion OTP. In future versions of the NLP-5x, this OTP may be expanded in steps of 32KW. This OTP memory is completely disabled when the -XM pin is pulled low. The Expansion OTP is mapped to both instruction space and data space so it can store both instructions and constant data.

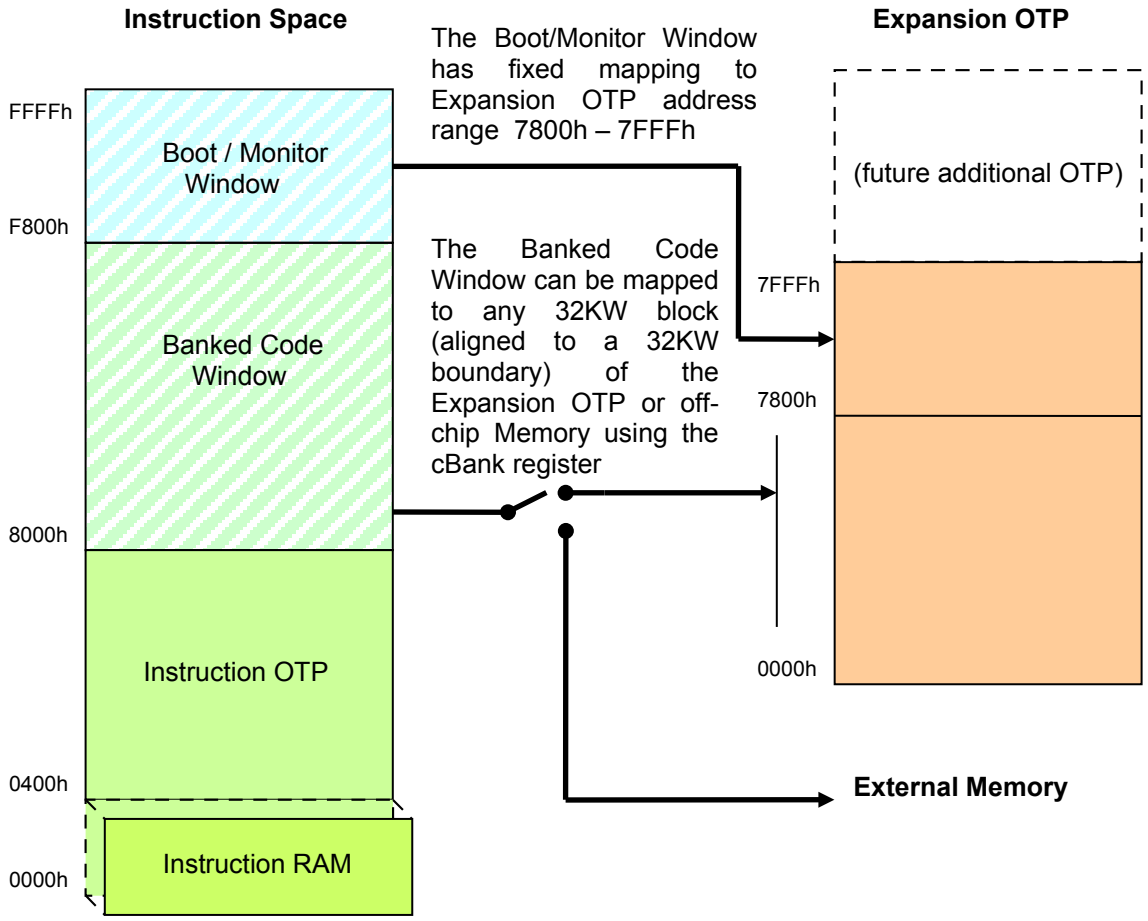
Finally, there is a small 256-word RAM accessed via the peripheral register space at addresses FD00h-FDFFh. The USB RAM is available to applications if not required for the USB interface.

An off-chip memory interface can be used for data and instructions as well. The pins associated with this interface are 23 address lines, A[22:0], 16 data lines, D[15:0], an active-low read strobe, -RD, an active-low write strobe, -WR, and 2 active low chip select outputs, -CS0 and -CS1. The address and data lines can be configured for general purpose I/O if not needed for external memory access. Separate wait states can be configured memory cycles with A[22] low vs. A[22] high.

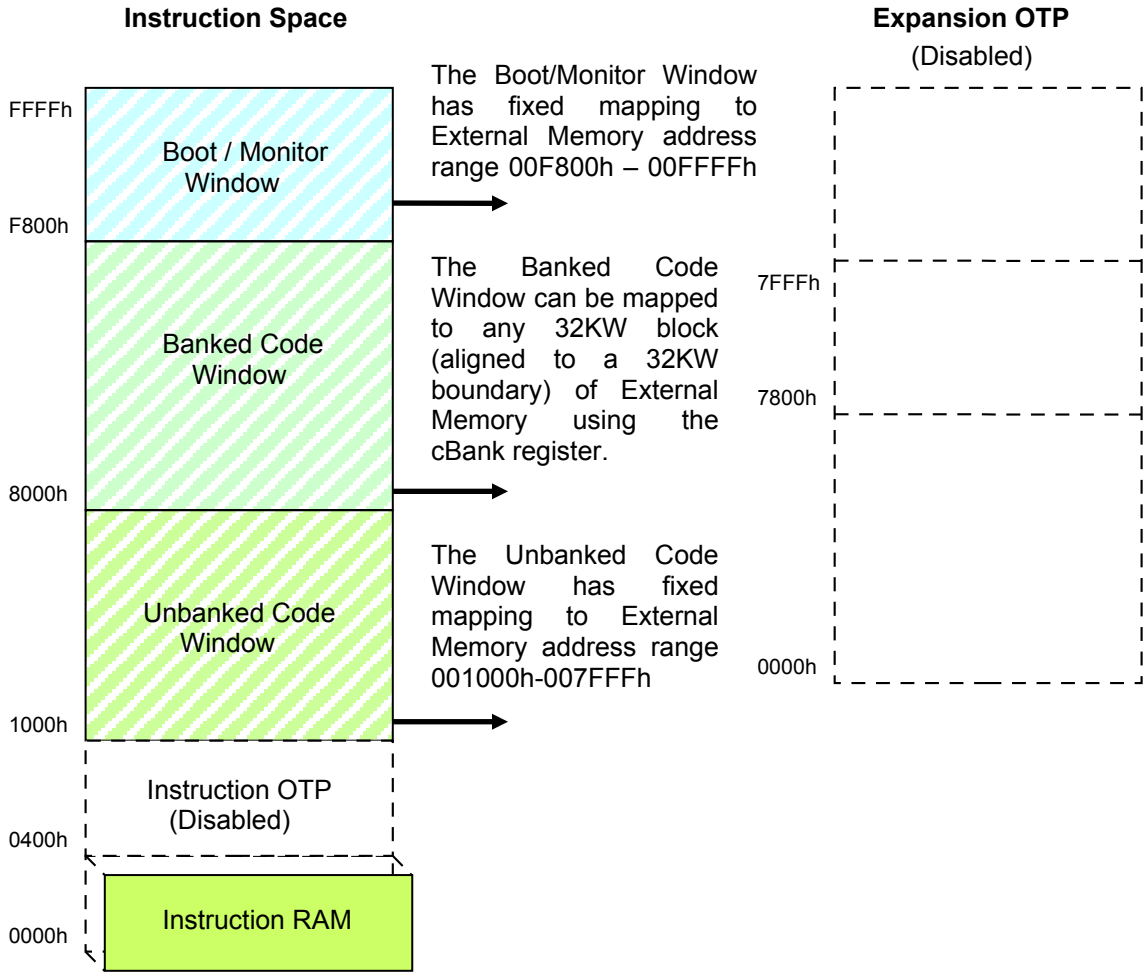
Instruction Space

The NLP-5x processor has a 16-bit instruction address, providing for 64KW of instruction space. The arrangement of instruction space is different depending on the state of the -XM pin, as shown in the following diagrams.

Instruction Space -XM = 1



Instruction Space -XM = 0



Unbanked Code Window
The lowest 32KW of instruction space is not banked. The first 1KW is assigned to the Instruction RAM.

If the -XM pin is high, addresses 0400h to 7FFFh are assigned to the on-chip Instruction OTP. The Instruction OTP address range 0000h to 03FFh is not normally accessible. However, if bit 11 of miscCtrl is set, this address range can be read.

If -XM is low, addresses 1000h to 7FFFh are directed to the external address bus, without banking (that is, not affected by the contents of the cBank register). The Instruction OTP is disabled and powered-down.

Banked Code Window
The code banking mechanism uses the cBank register to map instruction space logical addresses in the range 8000h-F7FFh to physical addresses in either the on-chip Expansion OTP or off-chip memory. A ***logical address*** is an address from the perspective of an executing program. A ***physical address*** is an address from the perspective of a memory device.

Code Bank Register (cBank) – 0xFC87

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
cx	0	0	0	0	0	0	0	cb7	cb6	cb5	cb4	cb3	cb2	cb1	cb0

cBank is reset to zero.

Using the bits CB[7:0] from the cBank register, instruction space logical addresses LA[15:0] in the range 8000h-F7FFh are mapped to physical addresses PA[22:0] as follows:

PA[14:0] = LA[14:0]
PA[22:15] = CB[7:0]

Thus, when cBank = 0000h, addresses 8000h-F7FFh are mapped to physical addresses 00000h-077FFh of the Expansion OTP. If cBank = 0001h, then addresses 8000h-F7FFh are mapped to physical addresses 08000h-0F7FFh of the Expansion OTP.

Note: in the first version of the NLP-5x, the NLP-5x1128, the Expansion OTP is only 32KW. The physical addresses PA[22:15] are not used and not connected to the Expansion OTP. Thus when CX=0 and -XM=1, the code bank bits are effectively ignored.

If the CX bit = 1 ***or*** the -XM pin = 0, then instruction space logical addresses (LA) in the range 8000-F7FF are mapped to physical addresses (PA) as above, but the read is directed off-chip. For example, if cBank = 8001h, then addresses 8000h-F7FFh are mapped to physical addresses 08000h-0F7FFh of off-chip memory. Note: for off-chip instruction memories, a 16-bit-wide memory device must be used.

Code in each bank will be compiled to start at a logical address 8000h regardless of the ultimate physical address. There will be a 2KW hole in each 32KW bank corresponding to the Boot/Monitor Window

Security Warning:
Instruction reads from off-chip can be enabled in software by setting the CX bit. When the -XM pin is high, the program in off-chip memory will be able to read all internal memories of the NLP-5x, so code and data cannot be protected. If security is important, the application should not set the CX bit. NOTE: Setting the CX bit should be rare, as applications should normally have more than enough instruction memory on chip.

Boot / Monitor Window

The uppermost 2KW (F800h – FFFFh) of instruction space is not banked, and this range is called the Boot/Monitor Window. Address F800h is the reset entry point for the NLP-5x, so boot code is located here. This memory range can also be used for a *monitor program* that assists in debugging activities.

If the -XM pin is high, instruction space reads from the address range F800h-FFFFh are always mapped to physical addresses 07800h-07FFFh of the on-chip Expansion OTP.

If the -XM pin is low, instruction space reads from the address range F800h-FFFFh are always mapped to physical addresses 00F800-00FFFF of the off-chip memory interface.

Instruction Space Memory Speed and Wait States

Instruction fetches from Instruction RAM can be as fast as the upper limit of the processor clock, CPUCLK, which is 80 MHz.

In contrast, the Instruction OTP can run up to 40 MHz without any wait states (access time of 25 nsec.). Above 40 MHz, in order to use the on-chip OTP memories, the NLP-5x must be configured in Turbo Mode, which is selected via bits [2:0] of clkSelect. In this mode, one wait state is inserted when reading from Instruction OTP.

The Expansion OTP also has a 25 nsec. access time. The Expansion Bus requires additional cycles for handshaking. A typical instruction fetch from Expansion OTP may take 3 cycles in normal mode, or 4 cycles in turbo mode.

Instruction fetches directed off-chip will use the wait states programmed in the External Memory Configuration Register (see xmConfig, pg. 25).

Hardware Breakpoint Registers

The NLP-5x supports a single hardware breakpoint. The contents of the breakpoint data register **bpData** will replace the instruction fetched from any instruction memory when the logical address and bank bits match the contents of the **bpAdrs** and **bpBank** registers, respectively, and if the breakpoint enable bit (bpBank[14]) is set.

Breakpoint Address Register (bpAdrs) – 0xFC8A

The bpAdrs register contains the 16-bit *logical address* of the hardware breakpoint.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
ba15	ba14	ba13	ba12	ba11	ba10	ba09	ba08	ba07	ba06	ba05	ba04	ba03	ba02	ba01	ba00

bpAdrs is reset to zero.

Breakpoint Bank Register (bpBank) – 0xFC8B

The bpBank register is compared against the current cBank bits during an instruction fetch to determine if a breakpoint should occur..

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
cbx	be	0	0	0	0	0	0	cb7	cb6	cb5	cb4	cb3	cb2	cb1	cb0

bpBank is reset to zero.

Bit[15] Breakpoint is in off-chip memory – ignored when the -XM pin is low.
Bit [14] Breakpoint Enable
Bit[7:0] Breakpoint Bank

Breakpoint Data Register (bpData) – 0xFC8C

The bpData register contains the breakpoint value.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
bd15	bd14	bd13	bd12	bd11	bd10	bd09	bd08	bd07	bd06	bd05	bd04	bd03	bd02	bd01	bd00

bpData is reset to zero.

Typically, for debugging, this register is set to 0AD9Eh, which is the instruction that sets the DEI interrupt: “bits %ireq, 14”.

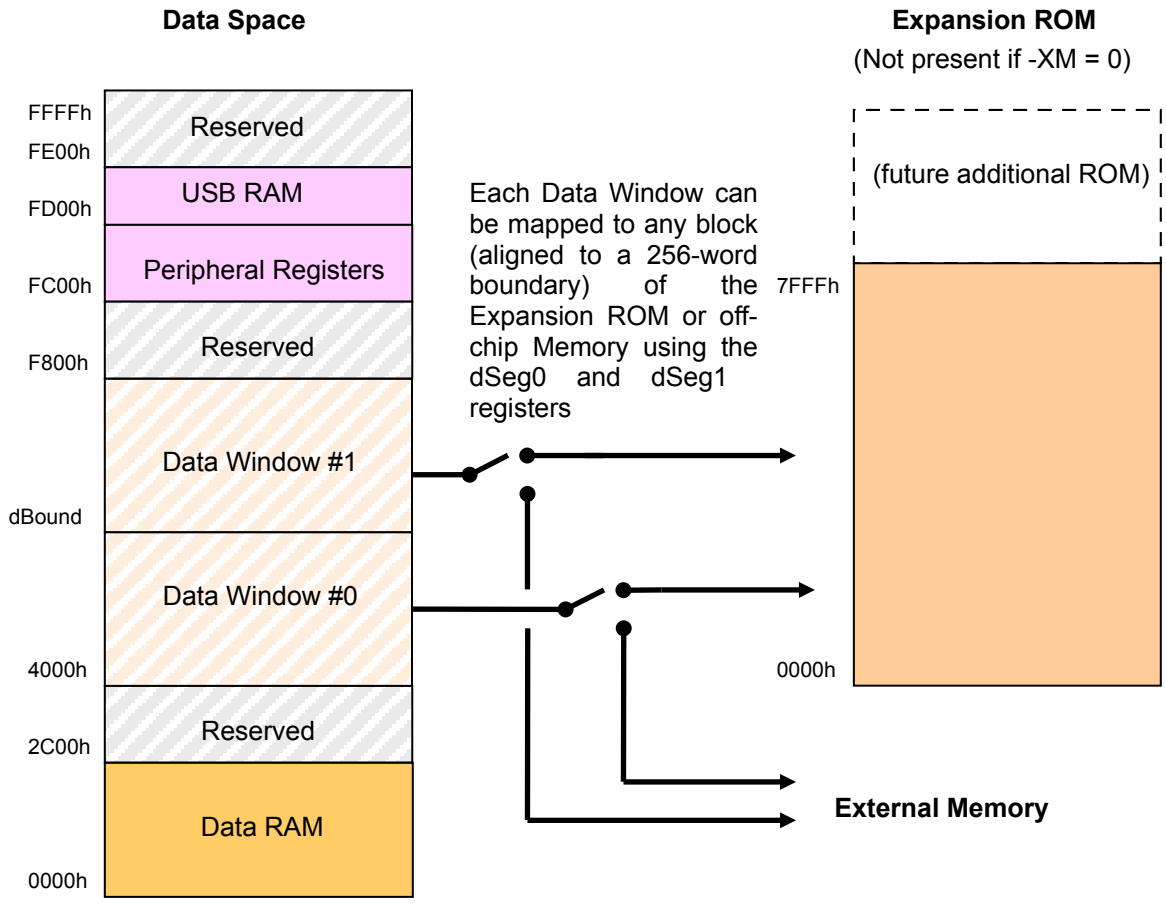
The breakpoint address is mapped to any of Instruction RAM, Instruction OTP, Expansion OTP, or off-chip memory in the same manner as instruction fetches use the logical address, the CX and CB[7:0] bits of the **cBank** register, and the -XM pin state.

If a match occurs between the instruction logical address and **bpAdrs**, and the current **cBank** matches **bpBank** (including the CX bit, unless -XM = 0, in which case the CX bit is ignored) and the breakpoint enable bit BE is set, then the instruction that is read from memory is replaced by the contents of the **bpData** register.

Data Space

The NLP-5x data space memory is accessed using load and store instructions with a 16-bit address range.

The Data Space of the NLP-5x is divided into a number of regions, as shown in the following diagram:



Data space reads and writes within the address range 0000h-2BFFh are directed to the 11KW Data RAM. The address range from 2C00h to 3FFFh is reserved for future expansion of the Data RAM.

All data space accesses at or above the address 4000h use the NLP-5x processor's Expansion Bus.

Data space from 4000h to F7FFh is divided into two *windows*. The dBound register is used to set the dividing line between the two windows. Each window can be mapped to an address range in the on-chip Expansion OTP or off-chip memory. The physical address range must be aligned to a page of 256 words.

The address range FC00h-FCFFh is assigned to peripheral registers. The address range FD00h-FDFFh is used by the RAM associated with the USB controller.

Data Segment 0 Register (dSeg0) – 0xFC85

The dSeg0 register maps addresses within the address range of window #0.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
dx0	seg0[14:0]														

dSeg0 is reset to zero.

DX0 high means Window #0 is directed to off-chip memories. If the -XM pin is zero, the state of DX0 does not matter because both windows are always directed off-chip.

Data Segment 1 Register (dSeg1) – 0xFC86

The dSeg1 register maps addresses within the address range of window #1.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
dx1	seg1[14:0]														

dSeg1 is reset to zero.

DX1 high means Window #1 is directed to off-chip memories. If the -XM pin is zero, the state of DX1 does not matter because both windows are always directed off-chip.

Data Segment Boundary Register (dBound) – 0xFC88

The dBound register defines the boundary between window #0 and data window #1.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
db15	db14	db13	db12	db11	db10	db09	db08	0	0	0	0	0	0	0	0

dBound is reset to 0

Bits [15:8] Defines bits [15:8] of the first address of window #1, and the last + 1 of window #0.
Bits [7:0] Ignored.

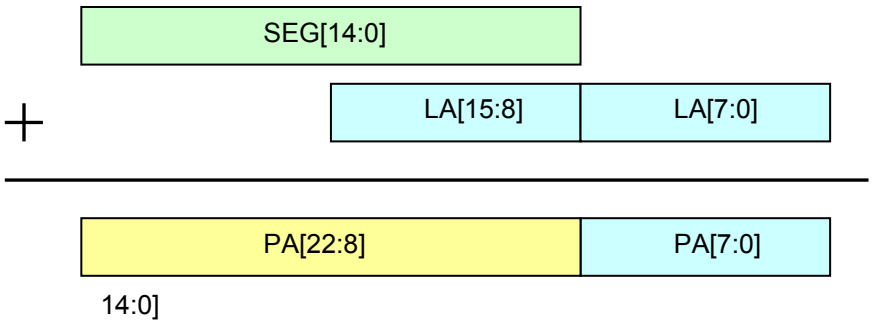
Note: the FluentChip™ technology library assumes that the dBound register is set to 0x8000. The library startup code initializes dBound to this value.

Data Segment Addressing

The data segment mechanism uses the dSeg0 and dSeg1 registers to map data space logical addresses in the range of either window #0 or window #1 to physical addresses in either on-chip Expansion OTP or off-chip memory. A ***logical address*** is an address from the perspective of an executing program. A ***physical address*** is an address from the perspective of a memory device.

A 23-bit physical address (PA[22:0]) for logical addresses (LA[15:0]) in the range of either of the two windows is generated using a 15-bit segment offset SEG[14:0] (from dSeg0[14:0] or dSeg1[14:0]) as follows:

PA[22:0] = (SEG[14:0] << 8) + LA[15:0]



If bit 15 of the segment register (dSeg0 for window #0, dSeg1 for window #1) is set to 1, or if the -XM pin is 0, the physical address is directed off-chip via the A[22:0] pins. Otherwise the physical address is directed to the Expansion OTP.

Note: in the first version of the NLP-5x, the NLP-5x1128, the Expansion OTP is only 32KW. The physical addresses PA[22:15] are ignored when accessing the Expansion OTP.

Data Space Memory Speed and Wait States

Data read and write operations from instruction RAM can be as fast as the upper limit of the processor clock, CPUCLK, which is 80 MHz.

The Expansion OTP has a 25 nsec access time. The Expansion Bus requires additional cycles for handshaking. A typical data read from Expansion OTP may take 3 cycles in normal mode, or 4 cycles in turbo mode. The same timing is used for writing and reading peripheral registers.

Data reads and writes directed off-chip will use the wait states programmed in the External Memory Configuration Register (see xmConfig, pg. 25).

External Memory Interface

Forty-four (44) pins are used to provide an interface between the NLP-5x and external parallel memories. External memories can be used for both instructions and data.

There are 23 address line outputs, A[22:0], addressing up to 8 MW without additional decoding. During reset the address pins will be configured as high impedance inputs with weak pull-up devices. The pins will stay in this state until the first off-chip memory cycle occurs, after which those pins that are configured as address lines (via the xmConfig register) rather than general purpose I/O will become outputs.

There are 16 data bus lines, D[15:0]. These pins are bi-directional: they are normally inputs except when there is an external write. By default, after reset, the data bus pins have weak pull-up devices (~260K ohm) that are enabled whenever the data bus is not driving as an output. The pull-up devices can be disabled via the xp0CtrlA/B registers.

There are 2 command strobes for reading from, and writing to, off-chip memory: active-low read strobe, -RD, and active-low write strobe, -WR. Both pins are high impedance with weak pull-ups during reset and outputs thereafter.

There are 2 active-low chip select outputs. By default, after reset, -CS0 will be active when A[22] = 0 , and -CS1 will be active when A[22] = 1. This mode is useful if there are two separate external memory chips. If bit 10 of xmConfig is zero, -CS0 will be active for all off-chip memory operations. This mode is useful if there is a single external memory that uses all 23 address lines. Both pins are high impedance with weak pull-ups during reset and outputs thereafter.

There is one input pin, -XM, to disable the on-chip Expansion OTP and Instruction OTP. This pin has a pull-up device that is enabled when reset is asserted low. The state of the XM_ pin state is latched shortly after reset is de-asserted, and if the -XM pin is low, the pull-up device is disabled to reduce current consumption. The latched state of the -XM pin can be read via bit 6 of register miscStatus.

External Memory Configuration Register (xmConfig) – 0xFC84

The xmConfig register provides means for configuring the external memory interface. The address and/or data bus may be configured for use as general purpose I/O if not needed for memory access. The auxiliary I/O pins are selected for I/O use via the xmConfig register, configured with the auxiliary port registers xp*n*CtrlA and xp*n*CtrlB, and read/written with the xp*n*In and xp*n*Out input and output registers. See the Auxiliary Ports section for configuration and use details.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
asel2	asel1	asel0	dhen	dlen	cs1en	rdcs1	csrd1	ws1[2]	ws1[1]	ws1[0]	rdcs0	csrd0	ws0[2]	ws0[1]	ws0[0]

xmConfig is reset to FFFFh.

Bits [15:13] ASEL[2:0] Define which address pins are used for memory or I/O:

0 0 0	all address lines are I/O	
0 0 1	A[22:8] is I/O	A[7:0] is address
0 1 0	A[22:16] is I/O	A[15:0] is address
0 1 1	A[22:18] is I/O	A[17:0] is address
1 0 0	A[22:20] is I/O	A[19:0] is address
1 0 1	A[22:21] is I/O	A[20:0] is address
1 1 0	A[22] is I/O	A[21:0] is address

1 1 1 no I/O A[22:0] is address

When used as I/O, address lines are mapped as follows:
A[15:0] are controlled by auxiliary port 1 (xp1).
A[22:16] are controlled by auxiliary port 2 (xp2).

Bit [12] DHEN Defines if D[15:8] are used for memory or I/O
0 D[15:8] are assigned to auxiliary port 0, xp0[15:8].
1 D[15:8] are used for memory access.

Bit [11] DLEN Defines if D[7:0] are used for memory or I/O
0 D[7:0] are assigned to auxiliary port 0, xp0[7:0].
1 D[7:0] are used for memory access.

Bit [10] CS1EN Defines if -CS1 is enabled.
0 -CS1 is never active. -CS0 is used for all memory accesses.
1 -CS1 is active for memory accesses with A[22] = 1,
-CS0 is active for memory accesses with A[22] = 0.

Bits [9:5] RDCD1, CSRD1, WS1[2:0] for accesses with A[22] = 1. See Timing section below.

Bits [4:0] RDCD0, CSRD0, WS0[2:0] for accesses with A[22] = 0. See Timing section below.

External Memory Read and Write Timing

Five bits are used to control the timing of off-chip memory access cycles. In normal (non-Turbo) mode the processor clock (CPUCLK) is the basis for off-chip memory access timing. In Turbo Mode, the processor clock divided-by-two (CPUCLK/2) is used instead. In Turbo Mode, the maximum memory clock rate is 40 MHz (25 nsec cycle time), and the shortest read cycle is two clocks (50 nsec). The shortest write cycle is 5 clocks, or 125 nsec.

There are two sets of these five bits: one set is used when A[22] is low, and the other is used when A[22] is high. This facilitates using A[22] to split external memory space among two devices where one is faster or slower than the other.

The five bits are:

RDCS Read High To Chip Select High Delay
1: there is a guaranteed clock cycle inserted after -RD goes high, before the address changes or chip select goes high.

0: address and chip select can change at the same time as -RD goes high.

This bit is ignored for write cycles.

CSRD Chip Select Low to Read Low Delay
1: there is a guaranteed clock cycle inserted after the address is presented and the chip select goes low, before -RD goes low.

0: -RD can go low at the same time as the address changes and chip select goes low.

This bit is ignored for write cycles.

WS[2:0] Defines the width of the -RD and -WR commands. At a minimum, these commands are two clocks wide (ie, 50 nsec with a memory clock of 40 MHz). The number of clocks is

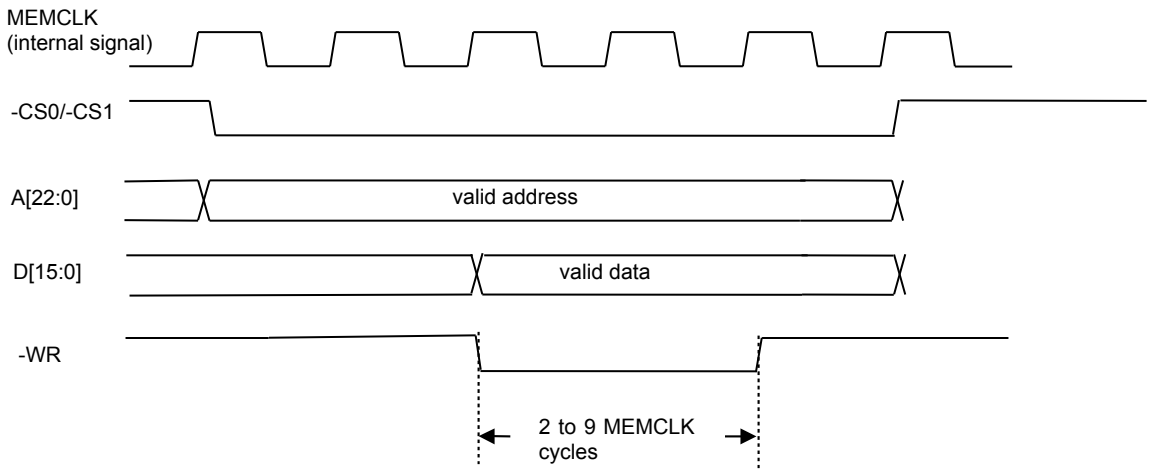
= 2 + WS[2:0]

memory clocks.

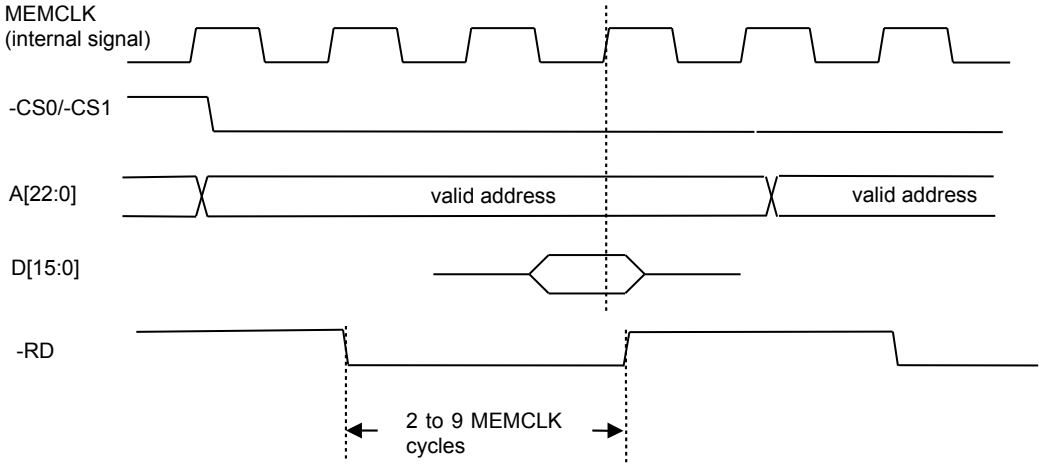
Write cycles start with the address and chip select output being asserted, followed by two memory clocks, followed by the write command being asserted for 2+WS[2:0] clocks, followed by one memory clock period with the address, chip select, and data held steady.

In the following diagrams, MEMCLK is CPUCLK/2 in Turbo Mode, and CPUCLK otherwise.

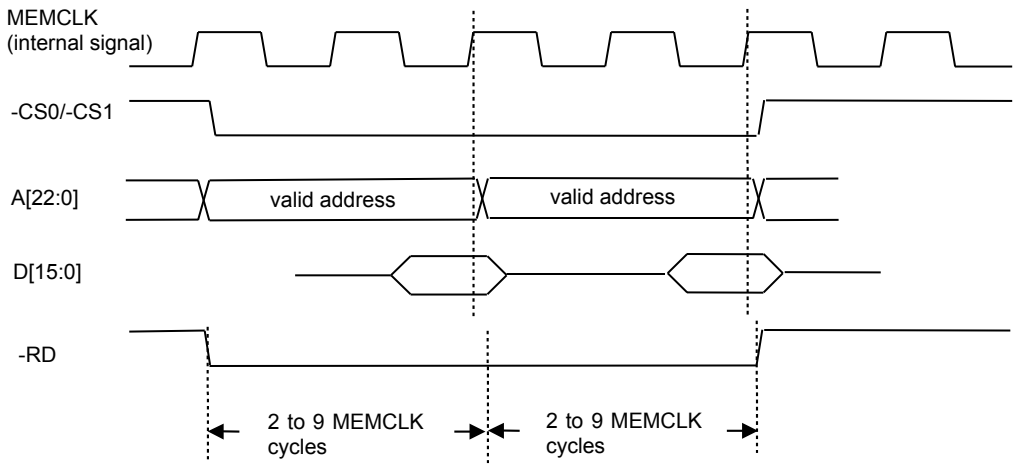
External Memory Write Cycle



Back-to-Back External Memory Read Cycles: CSRD = RDCS = 1



Back-to-Back External Memory Read Cycles: CSRD = RDCS = 0



External memory access time requirements for 72 MHz (Turbo Mode) or 36 MHz (non-Turbo Mode) operation as a function of the WS bits are listed in the table below. These are typical clock rates used by Fluent Chip technologies.

WS	# of MEMCLKs	Memory Tacc Required (ns)
0	2	37
1	3	65
2	4	93
3	5	121
4	6	148
5	7	176
6	8	204
7	9	232

-XM Pin

If the -XM pin is low ("External Mode") the internal Expansion OTP memory and the Instruction OTP memory will be disabled. All instruction fetches above or equal to address 1000h (the first 4KW) will be mapped off-chip regardless of the state of the CX bit in the cBank register. Instruction fetches below 400h (the first 1KW) will access the on-chip Instruction RAM. Instruction addresses between 400h and 0FFFh are unpopulated in this mode and will return unknown values.

All data space reads that would normally be directed to the Expansion OTP will go off-chip, regardless of the state of the DX bit in both the dSeg0 and dSeg1 segment registers.

This pin has a pull-up device that is enabled during any reset event, including power-on reset. The state of the -XM pin is latched shortly after reset is de-asserted, and, if at that moment the -XM pin is low, the pull-up device is disabled to reduce current consumption. This latched state of the -XM pin can be read via bit 6 of register miscStatus.

Reset and the External Memory Interface

While -RESET is held low, the external memory interface pins become high impedance pins with weak pull-up devices. After -RESET goes high, the control signals -CS0, -CS1, -RD and -WR become outputs that will be high until the first external memory access event. A[22:0] will remain high impedance with weak pull-up devices until the first external memory access event (or until configured by software as general purpose I/O). After the first memory access event those pins of A[22:0] that are not programmed to be general purpose I/O will become active outputs holding the last address read or written. D[15:0] are inputs with weak pull-ups except when there is a write event addressed to the external memory interface (unless configured as general purpose I/O).

NLP-5x Reset, Oscillators, Clocks, Power Management

Reset

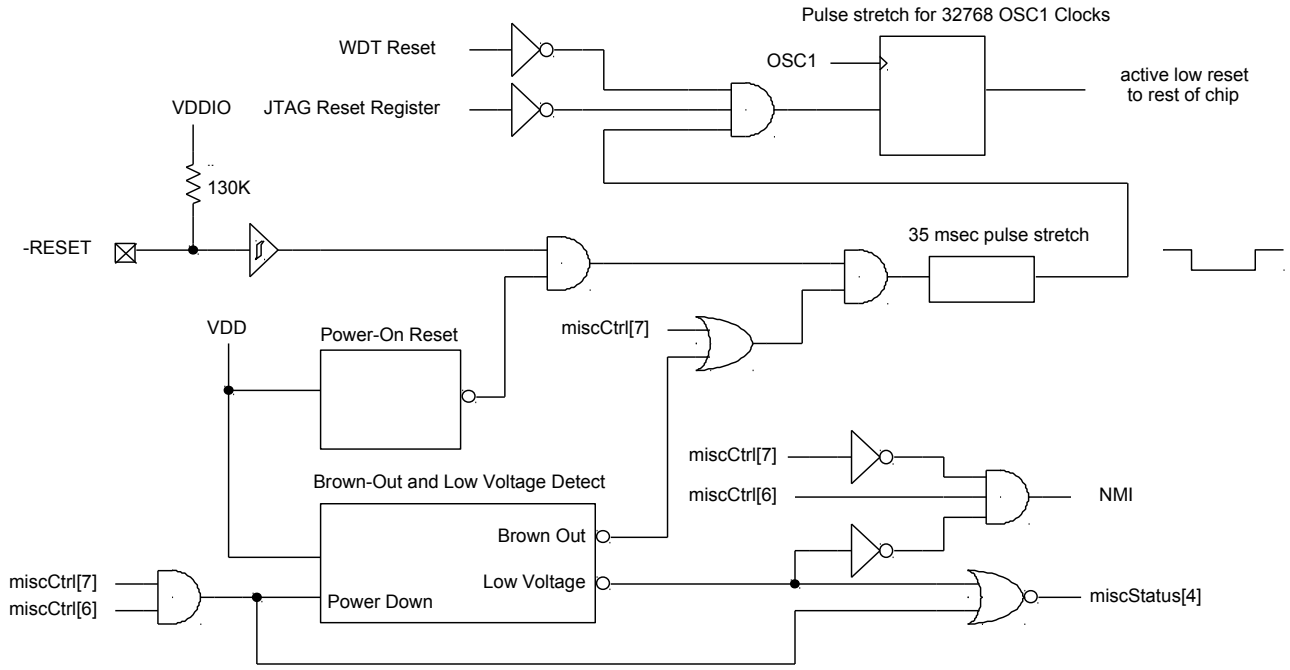
An external reset may be generated by a low voltage on the reset pin, -RESET. This pin is an active-low Schmitt trigger input with a weak pull-up (approximately 130Kohm). The reset signal from the -RESET pin is AND'd with the active-low output of the Power-On Reset circuit (POR).

The output of the Power-On Reset circuit will be low while the VDD voltage ramps up until it exceeds the POR upper threshold voltage (specified to be a maximum of 1.50V). Once the VDD voltage goes above the upper threshold voltage, the output of the POR will remain high until the VDD voltage is reduced below the POR lower threshold voltage, which is specified to be typically 0.90V.

Any type of reset event will enable the Brown-Out-Reset (BOR) circuit. This circuit has a more accurate reference and threshold voltage (1.50V +/- 60mV) than the POR. For the NLP-5x to come out of reset, VDD must be above the BOR voltage threshold. The BOR circuit also has an output for Low Voltage Detection (LVD). This circuit can be used to detect a low voltage condition with software polling or with an NMI interrupt. To reduce sleep current, the BOR/LVD circuit can be powered-down.

The combination of the above-named reset events passes through a detect-and-hold circuit that stretches the reset pulse to last approximately 35 milliseconds. This gives the OSC1 oscillator time to start up and settle. There are two other sources of reset: the Watch Dog Timer (WDT) and the JTAG Reset Register. After any reset event is de-asserted, the NLP-5x will be held in reset for a further period of 32768 OSC1 clock cycles.

Reset clears the global interrupt enable flag and begins execution at address F800h in the Boot/Monitor Window of Instruction Space.



Oscillators

There are three oscillators in NLP-5x, two of which are crystal-controlled. Each crystal-controlled oscillator is a tri-state inverter with a weak feedback device between the inverter output and the inverter input. The inverter and feedback devices are sized to operate at the nominal frequencies given below:

Oscillator #1	Pins XI1 and XO1	2.000-8.000 MHz, typical 4.000 MHz
Oscillator #2	Pins XI2 and XO2	32.768 KHz
Oscillator #3	(no pins, internal RC)	software adjustable over a wide range, 128 KHz +/- 5% typical after software calibration

OSC1 High Speed Oscillator and PLL

The high speed crystal oscillator uses pins XI1 and XO1 to connect to an external crystal or ceramic resonator. The rate is typically 4 MHz, but the oscillator is designed for rates in the range of 2 to 8 MHz. Whenever the high speed crystal oscillator is started, the output of the oscillator is blocked for 32768 clocks to allow for the rate to stabilize.

The high speed oscillator is enabled automatically by any reset event. It can be disabled (powered-down) by software clearing bit 0 of register hsoCtrl. It can also be disabled while in sleep mode if bit 0 of the sleep register is zero. In this case, the high speed oscillator will start automatically upon wake-up.

The high speed crystal oscillator is an input to a PLL function (Phase Locked Loop) with an internal VCO (Voltage Control Oscillator). The VCO rate can be programmed to be a multiple of the crystal rate with 9 possible multiplication factors, ranging from x8 to x40 in steps of 4. There are two outputs of the PLL function: the VCO clock divided-by-2, and the VCO clock divided-by-3.

The VCO clock divided-by-2 is typically the source of clock signal PLLCLK. PLLCLK is used for on-chip functions such as the ADC, PWM, and the DAC, and is typically the source of the processor clock, CPUCLK.

The VCO clock divided-by-3 can be selected as the clock to the USB function, and is also an optional source for the clock to the DAC.

Two typical VCO rates are 160 MHz and 144 MHz. The former is chosen to run the NLP-5x at its maximum specified processor speed of 80 MHz. The latter is chosen for USB applications: the processor clock is 72 MHz (VCO clock divided-by-2) and the USB clock is 48 MHz (VCO clock divided-by-3).

When the PLL function is not enabled (powered-down), or not selected via bit 2 of register hsoCtrl, or if the multiplier bits are all zero, the PLL is bypassed, meaning that PLLCLK is connected to the output of the high speed crystal oscillator.

The PLL will also be bypassed for 1024 crystal clocks whenever the hsoCtrl register is written on the assumption that the PLL multiplier or other control signals may have changed.

The PLL is also bypassed when the lock signal is low. The lock signal is an output of the PLL. The PLL will remain bypassed until the lock signal is high for 1024 OSC1 clocks in a row. Bit 7 of register hsoCtrl can be used to override the lock signal.

hsoCtrl - High Speed Oscillator Control Register (0xFC61)

15	14	13	12	11	10	09	08
lock	pc[5]	pc[4]	pllMul4	pllMul3	pllMul2	pllMul1	pllMul0
07	06	05	04	03	02	01	00
lockOverride	pc[3]	pc[2]	pc[1]	pc[0]	pllSelect	pllOn	hxoscOn

set to 0001h on reset

Bit [15] (R/O) Current state of PLL lock flag

Bit [14] PLL input PC[5]. Selects PLL LPF resistor value. It is recommended that this bit is always written with zero.

Bit [13] PLL input PC[4]. Selects PLL LPF capacitor value. It is recommended that this bit is always written with zero.

Bits [12:8] PLL multiplier

0 0 0 0 0	PLL is bypassed. PLLCLK is the high speed crystal output.
0 0 0 0 1	PLL VCO = input clock x 8
0 0 0 1 0	PLL VCO = input clock x 12
0 0 0 1 1	PLL VCO = input clock x 16
…	
0 1 0 0 0	PLL VCO = input clock x 36 (144 MHz if the crystal is 4 MHz)
0 1 0 0 1	PLL VCO = input clock x 40 (160 MHz if the crystal is 4 MHz)
other values	illegal

Bit [7] 0: PLL Lock output must be high for PLL output to be used.
1: PLL Lock output is ignored, but can be read via bit 15.

***Note**: it is recommended to set and leave bit 7 = 1 so that a rare, transient, lock loss does not cause the PLL rate to switch automatically to the OSC1 rate.*

Bits [6:5] PLL inputs PC[3:2]. Selects the VCO band.
Recommended values: PC[3] = 0, PC[2] = 1.

Bits [4:3] PLL inputs PC[1:0]. Selects the charge pump current. It is recommended that these bits are always written with zeros.

Bit [2] 0: PLL is bypassed. PLLCLK is the high speed crystal output. VCO divided-by-3 output is undefined.

1: Select the PLL output for PLLCLK. This will not take effect if the PLL lock signal is not asserted (or overridden via bit 7 of this register) or if the PLL is not enabled via bit 1 of this register.

Bit [1] 0: PLL is powered-down and bypassed. VCO divided-by-3 output is undefined.
1: Enable the PLL.

Bit [0] Enables the high speed crystal oscillator. This bit may be overridden while in sleep mode .

OSC2, OSC3 Low Speed Oscillators

There are two independent low speed oscillators. The low speed RC oscillator has software rate adjustment with 32 steps. Software adjustment provides enough control to tune the RC oscillator to a typical rate of 128 KHz within +/- 5%. The RC oscillator can be configured to divide its output by 4 to a typical rate of approximately 32 KHz. The RC oscillator will start immediately after being enabled. It also starts immediately after wakeup from sleep mode if it was disabled during sleep mode.

The low speed crystal oscillator uses pins XI2 and XO2 to connect to an external 32768 Hz crystal. When the low speed crystal oscillator is started, the output of the oscillator is blocked for 512 clocks to allow for the rate to stabilize. If OSC2 is not used in a system design, then XO2 should be left unconnected (open) but XI2 can either be left unconnected or tied to GND.

Either of the low speed oscillators can be used as the processor clock, CPUCLK. The clock select register (clkSelect) is used to select the CPUCLK source.

Timer #2 will use one of these two low speed oscillators, based on the state of bit [4] of lsoCtrl.

The LCD controller will use one of these two low speed oscillators, based on the state of bit [3] of lsoCtrl.

The WDT (watch dog timer) will use one of these two low speed oscillators, based on the state of bit [5] of lsoCtrl.

Both low speed oscillators are disabled by reset.

The operation of both low speed oscillators is controlled by register lsoCtrl, described next.

lsoCtrl - Low Speed Oscillator Control Register (0xFC60)

15	14	13	12	11	10	09	08
rcMon	lxMon	rcCap	rcFreq4	rcFreq3	rcFreq 2	rcFreq1	rcFreq0
07	06	05	04	03	02	01	00
		wdtSrc	t2Src	lcdSrc	rcdiv4	rcoscOn	lxoscOn

cleared to zero on reset

Bit [15] (R/O) Current state of RC oscillator output (prior to optional divide-by-4)

Bit [14] (R/O) Current state of Low Speed Crystal Oscillator

Bit [13] Capacitor trim input to RC block. This bit is normally programmed with zero.

Bits [12:8] RC oscillator frequency selection:
0 0 0 0 0 Highest Rate, 240 KHz +/- 40%
…
1 1 1 1 1 Lowest Rate, 20 KHz +/- 40%

Bit [5] Selects the source of the watch dog timer clock:
0 OSC3 - RC oscillator after optional divide-by-4
1 OSC2 - low speed crystal oscillator

Bit [4] Selects source of timer 2 clock:
0 OSC3 - RC oscillator after optional divide-by-4
1 OSC2 - low speed crystal oscillator

Bit [3] Selects source of LCD clock:
0 OSC3 - RC oscillator after optional divide-by-4
1 OSC2 - low speed crystal oscillator

Bit [2] RC oscillator divider option:
0 RC / 1
1 RC / 4

Bit [1] Enables OSC3 (RC oscillator). This bit may be overridden while in sleep mode.

Bit [0] Enables OSC2 (low speed crystal oscillator). This bit may be overridden while in sleep mode.

Clocks

This section describes some of the clock signals in the NLP-5x.

PLLCLK

If the PLL is "bypassed", PLLCLK is the same as the high speed oscillator output. Otherwise, it is the VCO (Voltage Controlled Oscillator) of the PLL divided-by-two.

Example: if bits [12:8] of register hsoCtrl are 01000 (8), then the VCO multiplication factor is x36. If a 4 MHz crystal is connected to OSC1, then the VCO clock rate will be 4 x 36 = 144 MHz. PLLCLK will be 72 MHz when the PLL is not bypassed and 4 MHz when it is bypassed.

PLLCLK is used for:

- ADC (Analog to Digital Converter)
- PWM (Pulse Width Modulator)
- SSP (Synchronous Serial Port)
- UART

PLLCLK can optionally be used for:

- Processor (CPUCLK)
- DAC (Digital to Analog Converter) -- the other choice is the PLL VCO divided-by-3
- USB (USBCLK) – the other choice is PLL VCO divided-by-3

USB Clock

The USB function requires a clock rate of 48 MHz. Bits [4:3] of the Clock Select Register (clkSelect) select which PLL output is used for the USB clock. There are three choices: VCO divided-by-2 (PLLCLK), VCO divided–by-3, or no clock.

Example: if bits [12:8] of register hsoCtrl are 01000 (8), then the VCO multiplication factor is x36. If a 4 MHz crystal is connected to OSC1, then the VCO clock rate will be 4 x 36 = 144 MHz. The VCO divided-by-3 output is 48 MHz and should be selected for the USB clock.

Processor Clock - CPUCLK

CPUCLK is used for the processor and also for the following functions:

- Infrared Interface
- Motor Controllers

There are 7 choices for CPUCLK, selected by bits [2:0] of the Clock Select Register (clkSelect), described next.

clkSelect – Clock Select Register 0xFC62

15	14	13	12	11	10	09	08
07	06	05	04	03	02	01	00
			usbvcod3	usbvcod2	cpuClk2	cpuClk1	cpuClk0

cleared to zero on reset.

Bit [4:3]
- 0 0 No USB clock
- 0 1 USB clock = PLL VCO / 2 (PLLCLK)
- 1 x USB clock = PLL VCO / 3

Bits [2:0]
- 0 0 0 CPUCLK is PLLCLK
- 0 0 1 CPUCLK is PLLCLK / 2
- 0 1 0 CPUCLK is PLLCLK / 4
- 0 1 1 CPUCLK is PLLCLK / 8
- 1 0 0 CPUCLK is PLLCLK, Turbo Mode
- 1 1 0 CPUCLK is from low speed crystal oscillator, OSC2
- 1 1 1 CPUCLK is from low speed RC oscillator, OSC3

If software changes the clock source, circuitry will prevent glitches.

Except in Turbo Mode, enabled by setting bits [2:0] to (1,0,0), or in External Mode, enabled by connecting pin -XM low, the CPUCLK must be configured for a rate less than or equal to 40 MHz. In Turbo Mode or External Mode, the CPUCLK rate can be configured up to 80 MHz.

Software should switch to Turbo Mode by setting bits [2:0] to (1,0,0) ***prior to*** configuring PLLCLK to be faster than 40 MHz.

Memory Clock – MEMCLK

This clock is used for the following functions:

- External Memory Interface Timing
- Timers #0, #1, and #3

MEMCLK is exactly the same as CPUCLK except in Turbo Mode.

In Turbo Mode, MEMCLK is CPUCLK divided-by-2.

Note: the NLP-5x emulator based on the VSI403LP chip will not divide CPUCLK by 2 in Turbo Mode for timers #0, #1, and #3. This difference between the emulator and the real NLP-5x chip must be accounted for in software when configuring any of these timers.

Watchdog Timer, Timer #2, LCD Clocks

The source clocks for these functions can be either of the two low speed oscillators. Bits 5, 4 and 3 of register IsoCtrl select the clock source for these functions.

Turbo Mode

The NLP-5x includes a mode of operation called Turbo Mode that allows the processor to run faster than if it was limited to the speed of the on-chip OTP memories. The Instruction OTP and Expansion OTP memories are limited to an access speed of 40 MHz. In Turbo Mode, the processor can run at twice this rate, or up to 80 MHz.

In Turbo Mode, a single wait state is inserted for each cycle that fetches data or instructions from either the Instruction OTP or the Expansion OTP.

The processor can gain an advantage in Turbo Mode in all these situations:

- It is executing a tight loop of 8 or fewer instructions entirely from its instruction cache.
- It is executing instructions stored in the Instruction RAM.
- When executing instructions from the Instruction OTP, most of the time this memory is read on every alternate cycle. This happens because this OTP is two words wide, so two instructions can be read at a time. This means that the processor can be expected to execute 2 instructions in 3 clocks, or an average rate of up to about 53 MHz. The actual rate is reduced by branch instructions.

In Turbo Mode, the clock signal MEMCLK is CPUCLK / 2 rather than CPUCLK. This clock is used by the External Memory Interface to determine the timing of off-chip read and write cycles. This clock is also used by Timer #0, Timer #1, and Timer #3.

In order to enable Turbo Mode, the clkSelect register, bits [2:0] should be set to (1,0,0) while the PLL is either bypassed or running at 40 MHz or slower. After switching to Turbo Mode, the PLL rate can be increased up to 80 MHz.

In the opposite situation of exiting Turbo Mode, the PLL rate should be reduced prior to changing bits [2:0] of clkSelect.

System Power Control and Wake Up

The DSP core used in the NLP-5x has power management features invoked via the lvl bits in the %smode register. The NLP-5x, however, does not use these bits for power management, and supports only the DSP core's normal operational mode (lvl=000). The NLP-5x implements a separate power management system that does not require any clock to be present as described below.

Sleep Mode

NLP-5x Sleep Mode is activated by writing to the sleep register, 0xFC64. The following subroutine shows an example. Sleep mode may be blocked as long as memory access cycles are in progress, so it is necessary to temporarily stop memory access cycles. A short loop with interrupts masked will not create any memory access cycles because the looped instructions are in the instruction cache.

***Note**: Correctly using sleep modes on the NLP-5x is quite complicated. Sensory urges developers to use the _Sleep() function provided in the FluentChip™ for NLP-5x library rather than attempting to write their own version. Source code for the library _Sleep() function is provided with the FluentChip™ for NLP-5x release for reference.*
In Very Low Power Sleep Mode, the on-chip Brown-Out-Reset function should be disabled to prevent false BOR events. These events can happen because the low-current, low-power regulator output voltage can vary much more than the normal high-current regulator. It is recommended that applications call _BorDisable() prior to _Sleep with SLEEP_VLPMODE selected.

```
/*
unsigned int _Sleep(unsigned int wakeMask, unsigned int sleepMode)
*/
        .global __Sleep

        .walign 2
        // r4 has wakeMask data
        // r5 has sleep register data
__Sleep:
        mov     r13, %imask             // save GIE bit
        bitc    %imask, 15              // clear GIE bit
        mov     %loop0, 4-1             // setup 4 loops
        lda     r6, wakeMask            // wakeMask register
        st      r4, r6                  // set wakeMask
        st      r5, r6, sleep-wakeMask  // send sleep request
        // sit in tight loop w/interrupts off to
        // temporarily block memory access cycles
SleepLoop:
        nop
        agn0    SleepLoop
        // if we get here,  we have awakened
        ld      r4, r6                  // read wake event
        mov     %imask, r13             // restore GIE bit
        ret
```

sleep - Sleep Mode Register (0xFC64)

write:

15	14	13	12	11	10	09	08
07	06	05	04	03	02	01	00
	vlpMode	-cmpPD	-afePD	ram_sleep_pd	-rcoscHalt	-lxoscHalt	-hsoscHalt

cleared to zero on reset**.**

Writing to this register will halt all clocks to the processor core and on-chip memories, until a wakeup event is detected, or reset. In addition, any of the oscillators on the chip can be halted for the duration of the sleep event. The PWM block will be powered-down automatically in sleep (outputs both low). The DAC drivers are not automatically changed to high-Z during sleep: software should explicitly disable the DAC prior to sleep if there is a DC load on the DAC outputs. GPIO pins (P0.0 – P0.15, P1.0 – P1.15, P2.0 – P2.7) configured as outputs will remain outputs in sleep mode and will continue to hold their pre-sleep output values.

Bit [6] Very Low Power Mode. A special low current regulator and reference generator is enabled during sleep. This only is active if bits 5, 4 and 0 are all zero.

Bit [5] 0: the comparator block is powered-down during sleep.
1: the comparator block can stay powered-up during sleep if one or both of the comparators is enabled.

Bit [4] 0: the ADC and preamp blocks are powered-down during sleep.
1: the ADC and preamp blocks can stay powered-up during sleep if their respective enable bits are set.

Bit [3] 1: The odd locations of the closely-coupled instruction RAM, and all locations of the data RAM are unpowered during sleep mode and their contents are lost.
0: The closely-coupled instruction and data RAM memories are powered during sleep mode and their contents are preserved.

Bit [2] If zero, the RC oscillator is stopped and powered-down during sleep, overriding bit 1 of the lsoCtrl register.

Bit [1] If zero, the low speed crystal oscillator is stopped and powered-down during sleep, overriding bit 0 of the lsoCtrl register.

Bit [0] If zero, the high speed oscillator will be stopped and powered-down during sleep, overriding bit 0 of the hsoCtrl register. The PLL will also be powered-down, overriding bit 1 of the hsoCtrl register. Warning: if this bit is set on the NLP-5x-DE emulator board, sleep may not be entered correctly if the CPU clock is selected to be faster than 36 MHz. It is recommended that the clkSelect register be used to slow down the CPU clock before entering sleep with bit 0 set when using the NLP-5x-DE emulator board.

The RC oscillator can restart immediately after wakeup. The high speed crystal oscillator output is blocked by an on-chip timer that typically lasts 35 msec. After this delay, the high speed oscillator and PLL combination is blocked by circuitry for a further 32768 cycles. The low speed crystal oscillator is blocked by circuitry for 512 cycles after wakeup.

Note: for minimum sleep current, the POR block should be powered down via bits [7:6] of miscCtrl.

read:

15	14	13	12	11	10	09	08
Low Voltage	JTAG Wake		USB Wake	IR	SSP	UART	USB Irq
07	06	05	04	03	02	01	00
		Timer 2	Comp B	Comp A	P0.6	P0.5	I/O Wake

Reading the sleep mode register returns a flag for each device that has seen a wakeup event since the sleep mode register was written.

Wakeup Events

The NLP-5x can be awakened from sleep mode by any of a number of events. The wakeMask register is used to select which events can cause the NLP-5x to exit sleep mode. Note: the JTAG wake event is not maskable.

Note: *Wakeup events are level sensitive, not edge sensitive.* For example, if the UART IRQ is asserted high prior to sleep mode, and bit 9 of wakeMask is set, the NLP-5x will immediately exit sleep mode.

wakeMask - Sleep Mode Wakeup Mask Register (0xFC63)

15	14	13	12	11	10	09	08
Low Voltage			USB Wake	IR	SSP	UART	USB Irq
07	06	05	04	03	02	01	00
		Timer 2	Comp B	Comp A	P0.6	P0.5	I/O Wake

cleared to zero on reset.

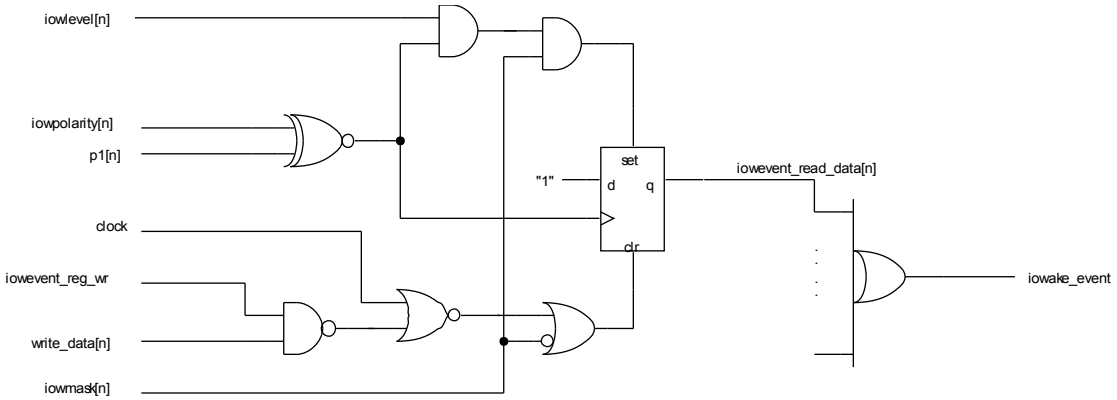
The IR, SSP and UART events require Osc #1 to be running during sleep mode. The Timer 2 event requires one of the low speed oscillators to be running during sleep mode. I/O Wake, USB Wake, the comparator events, and P0.6 and P0.5 events do not require a clock.

The USB Wake event is defined as the condition when the suspend status output of the USB controller is low, indicating that the USB controller is not in the suspend state.

The Low Voltage wakeup event is based on the low voltage detect output of the POR block (see diagram on page **31**). This is the same signal that can be read in bit 4 of miscStatus.

I/O Wake Event

Any of the port 1 GPIO pins can be configured to generate an I/O wake event. The I/O wake event can be used as an interrupt source, or as a method to wake from sleep mode. Four peripheral registers are used to configure the I/O wake event. The following diagram shows the wake event logic for a single port 1 pin.



Each GPIO pin of port 1 has a register that holds the I/O wake flag for that pin. All 16 registers are OR'd together to produce the I/O wake event interrupt request. The processor can read the iowEvent register to read the current state of these 16 registers to determine which pin(s) caused the I/O wake event. The processor can clear any register by writing a "1" to the corresponding bit of the iowEvent register. Writing a "0" to a register has no effect on the current contents of that register.

Each register has a mask bit. If the mask bit is clear, the I/O wake register for that pin is held low.

Each register has a level/edge mode selection bit. If the iowLevel bit is set for a given pin, that register is set high whenever the input pin state matches the iowPolarity bit for the same pin. If the iowLevel bit is clear, the register is set by the positive or negative edge, as determined by the iowPolarity bit.

I/O Wake Event Registers

There are four registers associated with the I/O Wake event function. Each bit of each register is mapped to the corresponding GPIO pins P1[15] to P1[0]. All four registers are cleared by reset.

iowEvent – I/O Wake Event Register 0xFC65

Write: if a "1" is written to a bit, the I/O wake register is cleared, if a "0" is written, there is no effect.
Any write to this register will force the I/O wake signal ("iowake_event" in the above diagram) low for the two processor clock cycles immediately following the write. If the I/O wake signal is high after a write to the iowEvent register, this mechanism will produce a low-to-high edge that can cause an interrupt request bit to be set in the %ireq register.

Read: returns the current state of the I/O wake registers.

iowMask – I/O Wake Mask Register 0xFC66

If the mask bit is zero, that port pin does not participate in I/O wakeup. Its register is held low.

iowPolarity – I/O Wake Polarity Register 0xFC67

This register determines the active state of the input pin. If set to one, the pin is active high, if zero, it is active low.

iowLevel – I/O Wake Level Sensitive Register 0xFC68

This register determines if the I/O wake event is level sensitive or edge sensitive. If set to one, the pin is level sensitive, if zero, it is edge sensitive.

Miscellaneous Interrupts, System Control

Beyond the I/O wake event interrupt request, there are six other sources of externally generated interrupt events that can be used for interrupts or wakeup events: P0.5 and P0.6 can be configured to latch an interrupt request on an edge of programmable polarity. P0.7 can be configured as an enable signal of the P0.6 edge event. Low Voltage Detect can be used to generate an NMI (non-maskable interrupt). The Watch Dog Timer (WDT) can be used to generate a reset.

P0.6, optionally enabled by P0.7, can be used to latch byte-wide data from P0.15-P0.8 into a register to implement a very simple host-peripheral interface.

The Miscellaneous Control register also can control the power down status of on-chip OTP (one-time-programmable) memories.

miscCtrl – Miscellaneous Control Register 0xFC69

15	14	13	12	11	10	09	08
		xotp_pd	ccotp_pd	iramrdis	wdt_mask	wdtPer1	wdtPer0
07	06	05	04	03	02	01	00
por_mode1	por_mode0	p07gate	p07pol	cbedge	caedge	p06edge	p05edge

cleared to zero on reset.

Bits [15:14] Reserved, software should write zeros to these bits.

Bit [13]
- 0 Slow OTP memory always powered-on except in sleep mode.
- 1 Slow OTP memory always powered-down.

Bit [12]
- 0 Closely-coupled OTP memory always powered-on except in sleep mode.
- 1 Closely-coupled OTP memory powered-down.

Bit [11]
- 1 Instruction RAM (0x0000-0x03FF) is disabled for reading but remains enabled for writing (if the sis bit of %smode is set). A read will access the fast code OTP data that is normally masked by the instruction RAM.
- 0 Instruction RAM is enabled for both reads and writes.

Bit [10]
- 1 Enable Watchdog Timer to generate a reset event.
- 0 WDT cannot generate a reset event. The WDT can still run and register an overflow which is readable via miscStatus[5].

Bits [9:8] The Watch Dog Timer (WDT) uses a clock from one of the two low speed oscillators, as determined by bit 5 of the lsoCtrl register.

0 0 WDT is disabled and the count reset to zero.
0 1 WDT is enabled, timeout is 4096 clocks of the selected WDT clock.
1 0 WDT is enabled, timeout is 32768 clocks of the selected WDT clock.
1 1 WDT is enabled, timeout is 262144 clocks of the selected WDT clock.

Bits [7:6] Power-On Reset Mode

0 0 POR block is powered up. The reset output is enabled. The low voltage detect output can be read via miscStatus[4] but does not generate an NMI event.

0 1 POR block is powered up, and the reset output is enabled. The low voltage detect output can generate an NMI event.

1 0 POR block is powered up, but the reset output is blocked. The low voltage detect output can be read via miscStatus[4] but does not generate an NMI event.

1 1 POR block is powered down. The reset output is blocked. The low voltage detect output is forced to zero.

Bit [5] P0.7 Function
0 P0.7 does not act as an enable signal for P0.6 edge events.
1 P0.7 acts as an enable for P0.6 edge events. This means that in order for P0.6 to generate an interrupt or latch the HPI register, P0.7 must be in the correct polarity as determined by bit 4 at the moment of the P0.6 edge.

Bit [4] P0.7 Polarity. Sets the active polarity for P0.7 to act as an enable signal for P0.6 edge events. If bit 5 of this register is 0, this bit is don't-care.
0 P0.7 is active low
1 P0.7 is active high

Bit [3] Comparator B interrupt source edge polarity.
0 Comparator B interrupt set by falling edge of comparator B output.
1 Comparator B interrupt set by rising edge of comparator B output.

Bit [2] Comparator A interrupt source edge polarity.
0 Comparator A interrupt set by falling edge of comparator A output.
1 Comparator A interrupt set by rising edge of comparator A output

Bit [1] P0.6 interrupt source edge polarity.
0 P0.6 interrupt set by falling edge of P0.6 state, optionally enabled by P0.7 if miscCtrl[5] is high.
1 P0.6 interrupt set by rising edge of P0.6 state, optionally enabled by P0.7 if miscCtrl[5] is high.

Bit [0] P0.5 interrupt source edge polarity.
0 P0.5 interrupt set by falling edge of P0.5 state.
1 P0.5 interrupt set by rising edge of P0.5 state.

miscStatus – Miscellaneous Status Register 0xFC6A

read*:*

15	14	13	12	11	10	09	08
				ver3	ver2	ver1	ver0
07	06	05	04	03	02	01	00
	-xm	wdt_overflow	lowVoltage	cb_ireq	ca_ireq	p06_ireq	p05_ireq

Bits [11:8] Chip revision number.

Bit [6] State of -XM pin as sensed after reset is deasserted.

Bit [5] The current state of the WDT overflow latch. If this bit is high, and miscCtrl[10] is high, a reset will be generated. Software can poll this bit to determine that the reset was caused by a WDT overflow.

Bit [4] The current state of the low voltage detect signal. High means the digital core voltage is below the low voltage detect threshold. If miscCtrl[7:6] = 01, then a change from low to high can cause an NMI interrupt.

Bit [3] Comparator B interrupt request latch. Set by positive or negative edge of the comparator B output. Polarity is determined by bit 3 of miscCtrl.

Bit [2] Comparator A interrupt request latch. Set by positive or negative edge of the comparator A output. Polarity is determined by bit 2 of miscCtrl.

Bit [1] P0[6] interrupt request latch. Set by positive or negative edge of the P0[6] pin. Polarity is determined by bit 1 of miscCtrl.

Bit [0] P0[5] interrupt request latch. Set by positive or negative edge of the P0[5] pin. Polarity is determined by bit 0 of miscCtrl.

write:

15	14	13	12	11	10	09	08
07	06	05	04	03	02	01	00
	wdt_reset_cnt	wdt_clr_ovf		cb_ireq_clr	ca_ireq_clr	p06_ireq_clr	p05_ireq_clr

Bit [6] Writing a “1” to this bit resets the WDT timer to zero.

Bit [5] Writing a “1” to this bit resets the WDT overflow register.

Bit [3] Writing a “1” to this bit resets the comparator B interrupt request register.

Bit [2] Writing a “1” to this bit resets the comparator A interrupt request register.

Bit [1] Writing a “1” to this bit resets the P0[6] interrupt request register.

Bit [0] Writing a “1” to this bit resets the P0[5] interrupt request register.

Host/Peripheral Interface Port

hpiData –Host/Peripheral Register 0xFC6B

read:

15	14	13	12	11	10	09	08
07	06	05	04	03	02	01	00
hprx_d7	hprx_d6	hprx_d5	hprx_d4	hprx_d3	hprx_d2	hprx_d1	hprx_d0

Read returns the contents of the Host/Peripheral data latch that is loaded from P0.15-P0.8 on a P0.6 edge, optionally enabled by the state of P0.7 if miscCtrl[5] is set.

Watch Dog Timer

The watch dog timer runs from one of the low speed oscillators, OSC2 or OSC3. Bit 5 of register lsoCtrl determines which oscillator is connected to the WDT:

lsoCtrl bit [5]	Selects the source of the watch dog timer clock:	
	0	OSC3 - RC oscillator after optional divide-by-4
	1	OSC2 - low speed crystal oscillator

Bits [9:8] of register miscCtrl determine whether the WDT is active and select the counter period. Bit 10 of miscCtrl determines whether a WDT timeout generates a reset event or just sets the WDT overflow flag.

miscCtrl bit [10]	1	Enable Watchdog Timer to generate a reset event.
	0	WDT cannot generate a reset event. The WDT can still run and register an overflow which is readable via miscStatus[5].
miscCtrl bits [9:8]	0 0	WDT is disabled and the count reset to zero.
	0 1	WDT is enabled, timeout is 4096 clocks of the selected WDT clock.
	1 0	WDT is enabled, timeout is 32768 clocks of the selected WDT clock.
	1 1	WDT is enabled, timeout is 262144 clocks of the selected WDT clock.

The WDT can be reset to zero by software by writing a 1 to bit 6 of register miscStatus.

The WDT overflow flag can be read by software reading bit 5 of register miscStatus.

If the WDT overflow flag is set after time-out, and bit 10 of miscCtrl is set, the WDT will cause a reset of the NLP-5x chip. This reset will act just like an external reset even on the –RESET pin, except that bit 5 of miscStatus will not be reset to zero. This allows startup code to poll bit 5 of miscStatus to determine if a reset event was caused by the WDT.

A WDT overflow can happen while the NLP-5x is in sleep mode. This will cause the chip to wake just as it would after an external reset event on the –RESET pin.

Interrupts

There are many sources of interrupt requests. The first group has fixed assignments to interrupt request bits in the %ireq register:

%ireq[5]	Timer #0 (internal to the NLP-5x Core)
%ireq[6]	Timer #1 (internal to the NLP-5x Core)
%ireq[13]	Software interrupt
%ireq[14]	Debug/Emulation non-maskable interrupt
%ireq[15]	Low Voltage Detect non-maskable interrupt

There is a second group of peripheral devices that have interrupt requests directly assigned to bits in the %ireq register, though they can also be selected as inputs to any of the 5 merged interrupts. These directly assigned interrupt pins cannot be used to wake the chip from sleep mode and they cannot be emulated with the VSI403LP part, so they will likely not be used except in rare situations. Most often, these interrupt sources are assigned to one of the 5 merged interrupt lines:

%ireq[7]	ADC FIFO
%ireq[8]	SSP
%ireq[9]	UART
%ireq[10]	USB
%ireq[11]	P0.5 Edge
%ireq[12]	P0.6 Edge

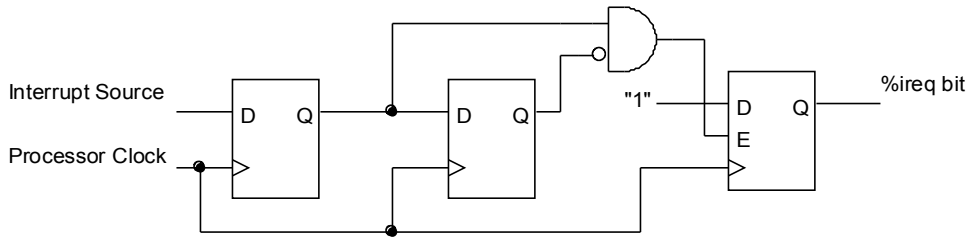
The final group of peripheral devices can be connected to one of 5 merged interrupt lines:

DAC FIFO
PWM FIFO
Comparator A
Comparator B
Timer #2
Timer #3
I/O Wake Event
IR (infra-red) receiver/transmitter
Motor control

Each interrupt request in the %ireq register is set by the corresponding interrupt request line changing from a low state (inactive) to a high state (active). It is this edge that sets the bit in the %ireq register. Even if the interrupt request line goes from high back to low, interrupt request bit in the %ireq register will remain high until the interrupt is acknowledged (i.e., the interrupt occurs) or explicitly cleared by software.

Each interrupt request line is sample and synchronized with the processor clock (CPUCLK) as part of edge detection, so each input must be both low and high for at least one full period of CPUCLK in order to be reliably detected.

The following diagram shows how an interrupt request bit can be set from an external event.



The interrupt source is latched into the first flip/flop on the rising edge of the processor clock (CPUCLK). If the current latched state of the interrupt source is high and the previous state was low, on the next clock the final flip/flop is enabled to load a 1. The output of this final flip/flop is the %ireq register bit. (Note that the %ireq bit can be set and cleared by software as well as by a hardware event; the software write mechanism is not shown in the diagram. However, this is not recommended for reasons given in the warning note in the description of the %ireq register).

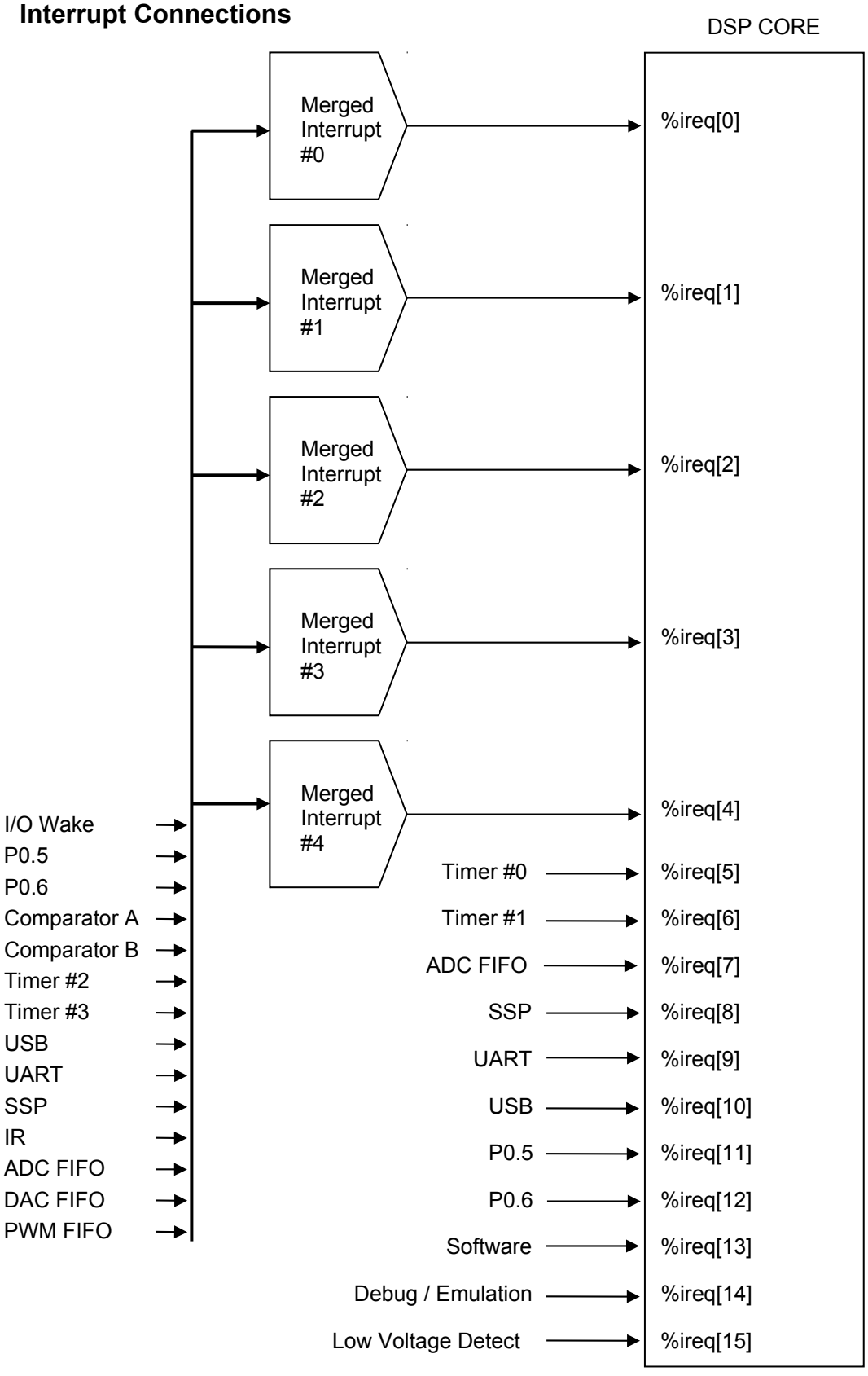
If an %ireq bit is set high and the corresponding %imask bit is set high and the global interrupt enable bit is set high, and the priority of the pending interrupt is equal to or greater than the current interrupt level, an interrupt will occur. Interrupts may be nested if software handles saving and restoring the hardware flags register (%hwflag).

A bit in the %ireq register that is set and causes an interrupt will be cleared automatically when the processor accepts the interrupt request and jumps to the entry point assigned to the interrupt.

If the corresponding mask register bit is clear, the %ireq bit will not cause an interrupt. However, it can be polled by reading the %ireq register.

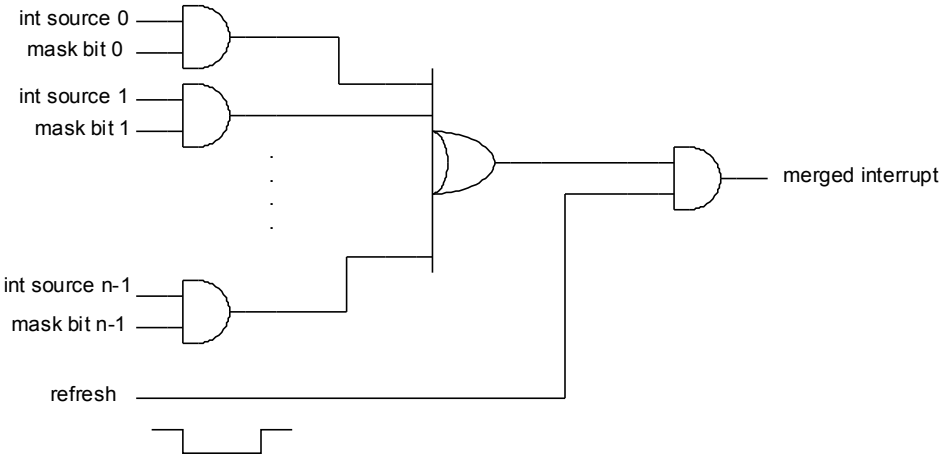
%ireq bits can be cleared by writing a 0 to the corresponding bit in the %ireq register (usually via the **bitc** instruction). %ireq bits can also be set by writing a 1 to the %ireq register via the **bits** instruction, creating a software-initiated interrupt. However, directly manipulating %ireq is not recommended for reasons given in the warning note in the description of the %ireq register.

Interrupt Connections



Merged Interrupts

There are more interrupt sources than there are interrupt pins on the VSI403LP chip which is used for emulation. For this reason the first 5 interrupt request bits, %ireq[4:0], are connected to interrupt merge blocks. Each merge block has 14 interrupt sources and 14 mask bits in a mask register. Those sources that have the mask bit set are OR'd together to a single merged interrupt. Software can generate a low going refresh pulse that lasts long enough to produce an edge inside the core in order to register an interrupt if any of the merged sources have a high input.



If only a single source is connected logically to a merged interrupt, then no special treatment is required for a merged interrupt in the interrupt service routine (ISR). In this case the single source is just passed through to processor. The ISR can assume that the interrupt request for the peripheral device is high and clear it by software command. Each peripheral device has its own method of clearing the interrupt request.

If more than one interrupt source is merged, then the interrupt service routine must poll each source that can be a part of a merged interrupt request and service the peripheral devices one-by-one. After servicing each device, the interrupt request for that device is cleared by software command. Each peripheral device has its own method of clearing the interrupt request. It is possible for a first device to assert an interrupt request and a second or third device to also assert interrupt requests while the ISR is handling the first device interrupt request. Depending on the times of arrival relative to the time of processing the devices in the ISR, it is possible that upon exiting a device may have its interrupt request asserted but no positive-going edge occurred and thus the interrupt was not registered in %ireq. To handle this case, before exiting the ISR, a 1 is written to a bit in the refresh register (miStatus) appropriate to the interrupt number in order to cause hardware to generate a low-going refresh pulse that lasts for 2 system clocks. This will generate a positive-going edge if any enabled device is asserting its interrupt request.

In another case it is possible that an edge *will* be generated, setting the %ireq bit high again, but after this the device is polled and processed. In this case the ISR will exit with the %ireq bit set even though the originating interrupt request is no longer asserted. In this rare case a second interrupt will happen. Each device will be polled but none will be processed and the ISR will exit without processing any device.

It is not recommended to use software to clear a bit in the %ireq register in the ISR (see the warning note in the description of the %ireq register).

Definition of Merged Interrupt Mask Registers miMask0, miMask1, miMask2, miMask3, and miMask4 (Registers 0xFC30 – 0xFC34)

15	14	13	12	11	10	09	08
PWM FIFO	DAC FIFO	ADC FIFO		IR	SSP	UART	USB
07	06	05	04	03	02	01	00
	Timer 3	Timer 2	Comp B	Comp A	P0.6	P0.5	I/O Wake

cleared to zero on reset.

There is a miMask register for each of the 5 merged interrupt request lines. Each interrupt source has a mask bit that must be set high in order to be OR'd into the merged interrupt request.

Merged Interrupt Source Status Register miStatus (0xFC35), (R/O)

15	14	13	12	11	10	09	08
PWM FIFO	DAC FIFO	ADC FIFO		IR	SSP	UART	USB
07	06	05	04	03	02	01	00
	Timer 3	Timer 2	Comp B	Comp A	P0.6	P0.5	I/O Wake

Reading this register returns the current state of the interrupt source status for each source that can be an input to a merged interrupt block.

Merged Interrupt Source Status Register miStatus (0xFC35), (W/O)

15	14	13	12	11	10	09	08
07	06	05	04	03	02	01	00
			refresh4	refresh3	refresh2	refresh1	refresh0

Bits [4:0] Write-only refresh command bits. For each merged interrupt block, if the corresponding bit is high when written, a 2-cycle active low pulse is generated to temporarily block the merged interrupt request and thereby produce a positive edge for the interrupt detection circuitry, if the merged interrupt was already high.

Interrupt Vector Table

For each interrupt, execution begins at a different address relative to the Interrupt Vector Table. The Interrupt Vector Table itself may be located in any of three locations, as shown in the following table. Please note that the Sensory FluentChip™ Technology Library assumes that the UVT bit of the %smode register is 0.

%smode.uvt	-XM Pin	Instruction Logical Address	Memory Space
0	LOW	0xF800	Off-Chip Memory
0	HIGH	0xF800	Internal Expansion OTP
1	n/a	0x0000	Instruction RAM

Event	Function	%smode.uvt=1, execution jumps to:	%smode.uvt=0 (default) execution jumps to:
%ireq.0	Merged Interrupt 0	0x0080	0xF880
%ireq.1	Merged Interrupt 1	0x0078	0xF878
%ireq.2	Merged Interrupt 2	0x0070	0xF870
%ireq.3	Merged Interrupt 3	0x0068	0xF868
%ireq.4	Merged Interrupt 4	0x0060	0xF860
%ireq.5	Timer 0 Interrupt	0x0058	0xF858
%ireq.6	Timer 1 Interrupt	0x0050	0xF850
%ireq.7	ADC (note 1)	0x0048	0xF848
%ireq.8	SSP (note 1)	0x0040	0xF840
%ireq.9	UART (note 1)	0x0038	0xF838
%ireq.10	USB (note 1)	0x0030	0xF830
%ireq.11	P0.5 Edge (note 1)	0x0028	0xF828
%ireq.12	P0.6 Edge (note 1)	0x0020	0xF820
%ireq.13	Software (note 1)	0x0018	0xF818
%ireq.14	Debug/Emulation	0x0010	0xF810
%ireq.15	Low Voltage Detect (note 1)	0x0008	0xF808
reset		n/a	0xF800

note 1: these interrupts cannot be emulated with the VSI403LP-based NLP-5x emulator.

Interrupt Mask Register (%imask)

The %imask register contains mask information for the 13 maskable interrupts supported by the NLP-5x. A cleared mask bit prevents the corresponding interrupt from being serviced. All bits in the %imask register are cleared on reset.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
gie	pgie	SW	P0.6	P0.5	USB	UART	SSP	ADC	T1	T0	M4	M3	M2	M1	M0

Bit [15] global interrupt enable. This bit is automatically cleared when an interrupt service routine is entered, and is restored from the contents of the pgie field by the reti instruction. Set this bit within an interrupt service routine to nest interrupts. When set, this bit enables all unmasked interrupts. When cleared, this bit disables all interrupts except NMI and the DEI.
Bit [14] This bit contains the original value of gie when executing an interrupt service routine. Execution of the reti instruction restores the value in pgie in the gie bit.
Bit [13] Software interrupt
Bit [12] P0.6 edge interrupt
Bit [11] P0.5 edge interrupt
Bit [10] USB controller interrupt
Bit [9] UART interrupt
Bit [8] SSP interrupt
Bit [7] ADC FIFO interrupt
Bit [6] Timer #1 Interrupt
Bit [5] Timer #0 Interrupt
Bit [4] Merged Interrupt #4
Bit [3] Merged Interrupt #3
Bit [2] Merged Interrupt #2
Bit [1] Merged Interrupt #1
Bit [0] Merged Interrupt #0

Interrupt Request Register (%ireq)

The %ireq control register shows the pending interrupts for all sources. Any of these bits may be set through software to create a user trap. All bits are sticky in the sense that a pending interrupt status is only cleared when that interrupt is serviced or the bit is explicitly cleared by software. This register is cleared at reset.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
NMI	DEI	SW	P0.6	P0.5	USB	UART	SSP	ADC	T1	T0	M4	M3	M2	M1	M0

Bit [15] Non-maskable Interrupt request used by Low Voltage Detect
Bit [14] Device emulation interrupt request
Bit [13] Software interrupt request
Bit [12] P0.6 edge interrupt request
Bit [11] P0.5 edge interrupt request
Bit [10] USB controller interrupt request
Bit [9] UART interrupt request
Bit [8] SSP interrupt request
Bit [7] ADC FIFO interrupt request
Bit [6] Timer #1 Interrupt request
Bit [5] Timer #0 Interrupt request
Bit [4] Merged Interrupt #4 request
Bit [3] Merged Interrupt #3 request

Bit [2] Merged Interrupt #2 request
Bit [1] Merged Interrupt #1 request
Bit [0] Merged Interrupt #0 request

Important Warning: The VSI403LP chip that is used on the NLP-DE development board has a problem related to the %ireq register that is not present in the actual NLP chip. On the VSI403LP, using a bitc/bits instruction to clear or set a bit in the %ireq register can clear *another* bit in this register if the bitc/bits happens at just the wrong moment. Interrupt registration in the %ireq register is edge-sensitive, i.e., based on a positive-going edge of the interrupt request signal from a device. For the interrupt sources other that T0/T1, software must clear the interrupt request for a device in order to generate a new positive-going edge (or generate an edge by writing to the miStatus register). Usually this is done in the interrupt service routine. Thus if an interrupt registration is missed, the interrupt never occurs, and the interrupt request from the device is not reset, and no further interrupts will be registered for that device. T0 and T1 will generate an edge at each time period, so a bitc/bits can cause an interrupt to be missed but when the next time period elapses they will be able to register a new interrupt request.

For this reason it is recommended that applications avoid using bitc and bits instructions with the %ireq register. If it is necessary to do so, disable interrupts before and after the bitc/bits instruction, and write 0x1f to the miStatus register after the bitc/bits but before enabling interrupts. This will cause any merged interrupt with a high interrupt request to be re-registerd in the %ireq register if it was cleared in error by bitc/bits. It is still possible for T0 and T1 register interrupt request to be cleared, in which case a single T0 and/or T1 interrupt may be missed. This method assumes that only T0, T1 and the merged interrupts are being used by the application. Since these are the only interrupts supported by the NLP-DE board, this assumption is usually valid.

Interrupt Priority Register 0 (%ip0)
The %ip0 register contains interrupt priority level and processor execution priority level information. The user may write to any field of this register. User-defined priorities are given values of 0b00 to 0b11, with 0b11 being the highest user-defined priority and 0b00 the lowest. This register contains 0x0 at reset.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
iepl		pepl		software		P0.6		P0.5		USB		UART		SSP	

Bits [15:14] Current execution priority level. Interrupts with LOWER priority will not be serviced. Interrupts with HIGHER or EQUAL priority will be serviced.
Bits [13:12] Previous execution priority. Written from bits 15-14 when an interrupt is taken.
Bits [11:10] Software interrupt priority level
Bits [9:8] P0.6 edge detect interrupt priority level
Bits [7:6] P0.5 edge detect interrupt priority level
Bits [5:4] USB controller interrupt priority level
Bits [3:2] UART interrupt priority level
Bits [1:0] SSP interrupt priority level

Interrupt Priority Register 1 (%ip1)
The %ip1 register sets interrupt priority level and controls interrupt behavior of the timer and external interrupts. The user may write to any field of this register. User-defined priorities are given values of 0b00 to 0b11, with 0b11 being the highest user-defined priority and 0b00 the lowest. This register contains 0x0 at reset.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
ADC FIFO		Timer 1		Timer 0		M4		M3		M2		M1		M0	

Bits [15:14] ADC FIFO
Bits [13:12] Timer #1 interrupt priority level

Bits [11:10]	Timer #0 interrupt priority level
Bits [9:8]	Merged interrupt #4 priority level
Bits [7:6]	Merged interrupt #3 priority level
Bits [5:4]	Merged interrupt #2 priority level
Bits [3:2]	Merged interrupt #1 priority level
Bits [1:0]	Merged interrupt #0 priority level

General Purpose I/O and Timers

Ports 0, 1, 2

The NLP-5x has 40 dedicated general purpose I/O pins (P0.0-P0.15, P1.0-P1.15, P2.0-P2.7). Each pin can be programmed as an input with weak pull-up (~260K equivalent device), input with strong pull-up (~11.5K ohm equivalent device), input without pull-up, or as an output. This is accomplished by having 2 bits of configuration for each GPIO pin. After reset, P0[15:0], P1[15:8], and P2[7:0] pins are configured as inputs with weak pull-ups. After reset, pins P1[7:0] are configured as analog input pins with no pull-ups, with the digital input buffers disabled (see register lineCtrl). Any of the P1 pins can be used as I/O Wakeup events. Any of the eight P1[7:0] pins can be selected as comparator inputs or as inputs for line-level ADC conversion.

Peripheral Registers for Port 0, Port 1 and Port 2

Name	Address	R/W	Reset State	Function
p0Out	0xFC00	R/W	0x0000	P0[15:0] output register
p0In	0xFC01	R		P0[15:0] input
p0CtrlA	0xFC02	R/W	0x0000	P0[15:0] control register A
p0CtrlB	0xFC03	R/W	0x0000	P0[15:0] control register B
p1Out	0xFC04	R/W	0x0000	P1[15:0] output register
p1In	0xFC05	R		P1[15:0] input
p1CtrlA	0xFC06	R/W	0x0000	P1[15:0] control register A
p1CtrlB	0xFC07	R/W	0x00FF	P1[15:0] control register B
p2Out	0xFC38	R/W	0x--00	P2[7:0] output register
p2In	0xFC39	R		P2[7:0] input. Bits [15:8] will read zero
p2CtrlA	0xFC3A	R/W	0x--00	P2[7:0] control register A
p2CtrlB	0xFC3B	R/W	0x--00	P2[7:0] control register B

Note: Port 1 bits [7:0] are reset to be analog inputs without internal pull-ups, with the digital input buffers disabled via register lineCtrl. All other GPIO pins are reset to be digital inputs with weak pull-ups.

Note:
Pins P0.15-P0.8 are Schmitt Trigger inputs. The remaining GPIO pins have normal input buffers.

The control registers A and B together control the function of the general purpose ports:

B	A	Function
0	0	Input - Weak Pull-up
0	1	Input - Strong Pull-up
1	0	Input - No pull-up
1	1	Output

For example, if register 0xFC03 bit 4 is set high, and register 0xFC02 bit 4 is low, then pin P0.4 is an input without a pull-up device.

In addition to their ordinary IO function described here, the P0, P1 and P2 pins all have alternate functions, such as LCD drivers, analog functions, serial ports, Host/Peripheral Interface, etc. These alternate functions are described throughout this document.

Auxiliary Ports XP0, XP1, XP2

If the external address bus A[22:0] and data bus D[15:0] are not needed in an application, these pins may be configured as auxiliary IO, bringing the total general purpose IO to 40+16+23=79. The auxiliary IO pins are selected with the xmConfig register, configured with the xp*n*CtrlA and xp*n*CtrlB registers, and read/written with the xp*n*In and xp*n*Out input and output registers. These auxiliary IO pins provide no wakeup, comparator input, or other such features, and they cannot be configured with strong pull-ups.

The mapping between auxiliary ports and the data/address bus is:

Port	**Pins**
XP0[15:0]	D[15:0]
XP1[15:0]	A[15:0]
XP2[6:0]	A[22:16]

Peripheral Registers for Auxiliary Ports XP0, XP1 and XP2

Name	Address	R/W	Reset State	Function
xp0Out	0xFCA0	R/W	0x0000	XP0[15:0] output register
xp0In	0xFCA1	R		XP0[15:0] input
xp0CtrlA	0xFCA2	R/W	0x0000	XP0[15:0] control register A
xp0CtrlB	0xFCA3	R/W	0x0000	XP0[15:0] control register B
xp1Out	0xFCA4	R/W	0x0000	XP1[15:0] output register
xp1In	0xFCA5	R		XP1[15:0] input
xp1CtrlA	0xFCA6	R/W	0x0000	XP1[15:0] control register A
xp1CtrlB	0xFCA7	R/W	0x0000	XP1[15:0] control register B
xp2Out	0xFCA8	R/W	0x--00	XP2[6:0] output register
xp2In	0xFCA9	R		XP2[6:0] input. Bits [15:7] will read zero
xp2CtrlA	0xFCAA	R/W	0x--00	XP2[7:0] control register A
xp2CtrlB	0xFCAB	R/W	0x--00	XP2[7:0] control register B

The control registers xp*n*CtrlA and xp*n*CtrlB together control the function of the auxiliary purpose ports:

B	A	Function
0	0	Input - Weak Pull-up
0	1	(unused) – strong pull-up not available
1	0	Input - No pull-up
1	1	Output

The auxiliary port configuration and input/output registers are enabled selectively by the xmConfig register as shown in the details of that register (see pg. 25). Any bit not enabled as auxiliary IO in the xmConfig register is don't-care in the auxiliary port registers.

Alternate I/O Functions

Each of P0, P1, and P2 can have alternate functions.

Port 2 Alternate Functions

P2 pins can be used for serial port functions. Each device has configuration registers that can take control of a group of P2 pins away from their GPIO function as programmed in p2CtrlA and p2CtrlB registers.

Pin	Name	Function	Configuration Override
P2.7	TXD	UART transmit	Becomes an output when UART is enabled regardless of the state of p2CtrlA[7] and p2CtrlB[7]
P2.6	RXD	UART receive	No override. If the UART is enabled, P2.6 should be configured as an input with or without a pull-up.
P2.5	TXIR	IR transmit	output
P2.4	RXIR	IR receive	input, no pullup
P2.3	MOSI	SSP *1 master-out/slave-in	Becomes an output, regardless of the state of p2CtrlA[3] and p2CtrlB[3], when SSP is enabled in SPI master mode or serial ROM mode. For other SSP modes, P2.3 must be configured as an input with or without a pull-up.
P2.2	MISO	SSP master-in/slave-out	Becomes an output, regardless of the state of p2CtrlA[2] and p2CtrlB[2], when SSP is enabled in I2S mode, also if SPI slave mode when the SS_ pin is low. In SPI master mode P2.2 should be configured as an input with or without a pull-up. Not used in serial ROM mode.
P2.1	SCLK	SSP serial clock	Becomes an output, regardless of the state of p2CtrlA[1] and p2CtrlB[1], when SSP is enabled in SPI master mode or serial ROM mode. For other modes, P2.1 must be configured as an input with or without a pull-up.
P2.0	SS_	SSP slave select	Becomes an output, regardless of the state of p2CtrlA[0] and p2CtrlB[0], when SSP is enabled in SPI master mode and sspCtrl[4] is high, or in serial ROM mode. For I2S mode and SPI slave mode P2.0 must be configured as an input with or without a pull-up.

Note *1: SSP stands for Synchronous Serial Port.

Port 1 Analog Functions

P1[7:0] pins can have optional analog functions. The digital input buffer for each of these pins can be enabled or disabled via the lineCtrl register. The digital input buffers are disabled by default after reset. Each of P1[7:0] can be selected as a source for the third A/D channel based on the state of lineCtrl[14:12]. Each of P1[7:0] can be an input to either the positive or negative terminal of either of the two comparators, based on the state of cmpACtrl and cmpBCtrl registers. Typically, software will configure those pins used for analog functions as inputs without pull-ups, with the digital input buffers disabled to reduce current consumption.

LCD Functions

Any or all of port 0 and port 1 pins can be used to directly drive an LCD display. If a LCD mode other than static drive is selected, P1.15 and P1.14 will be connected to an off-chip resistor divider to generate intermediate voltages for the LCD driver. In static mode, P1.15 is the common output. In other modes, two, three, or four of P1.13-P1.10 will become common outputs, depending on the number of common pins configured. All remaining P1 and P0 pins can be individually configured as LCD segment outputs.

I/O Wakeup Functions

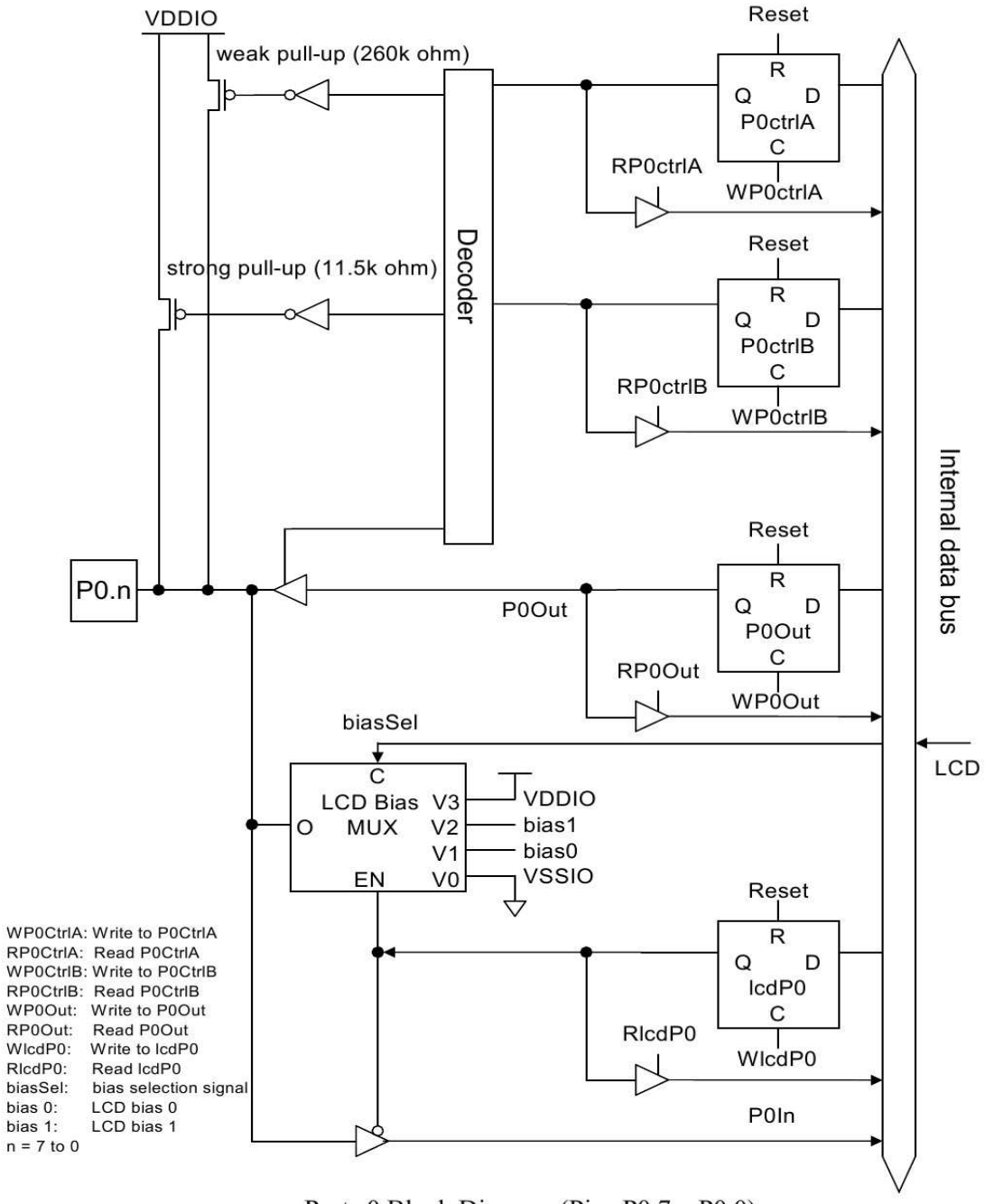
Any of P1 pins can be configured to generate an I/O wakeup interrupt event.

Port 0 Alternate Functions

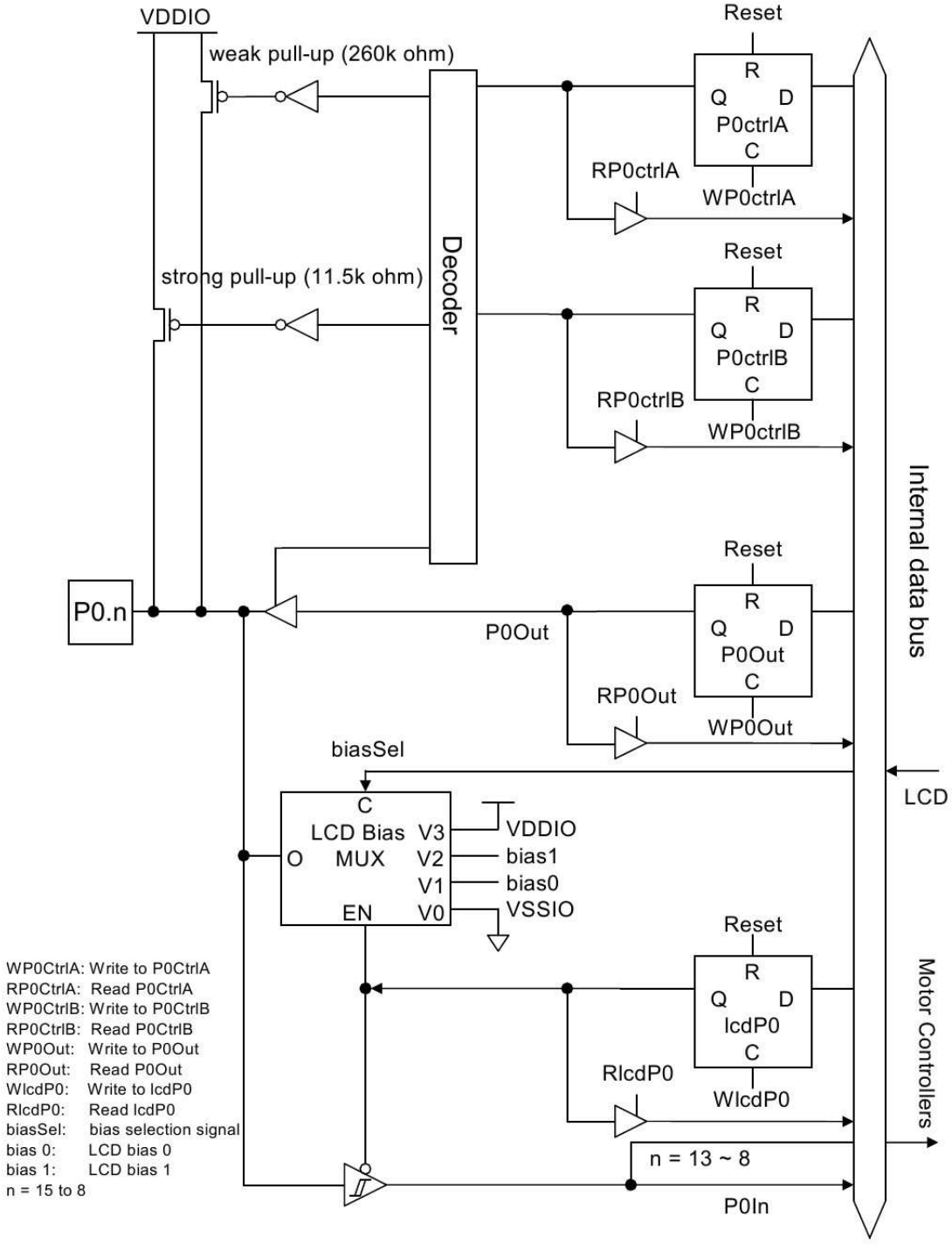
Pin	Name	Function
P0.15-P0.8	HPIDATA	Input Data for write to HPI register
P0.7	HPIEN	Optional enable for P0.6 edge event
P0.6	HPIWR	P0.6 edge interrupt and write strobe for HPI register
P0.5	---, T3CLK	P0.5 edge interrupt, timer #3 clock
P0.4	T3GATE	timer #3 gate

In each case, software must use the p0CtrlA and p0CtrlB registers to set the direction and pull-up state of each pin.

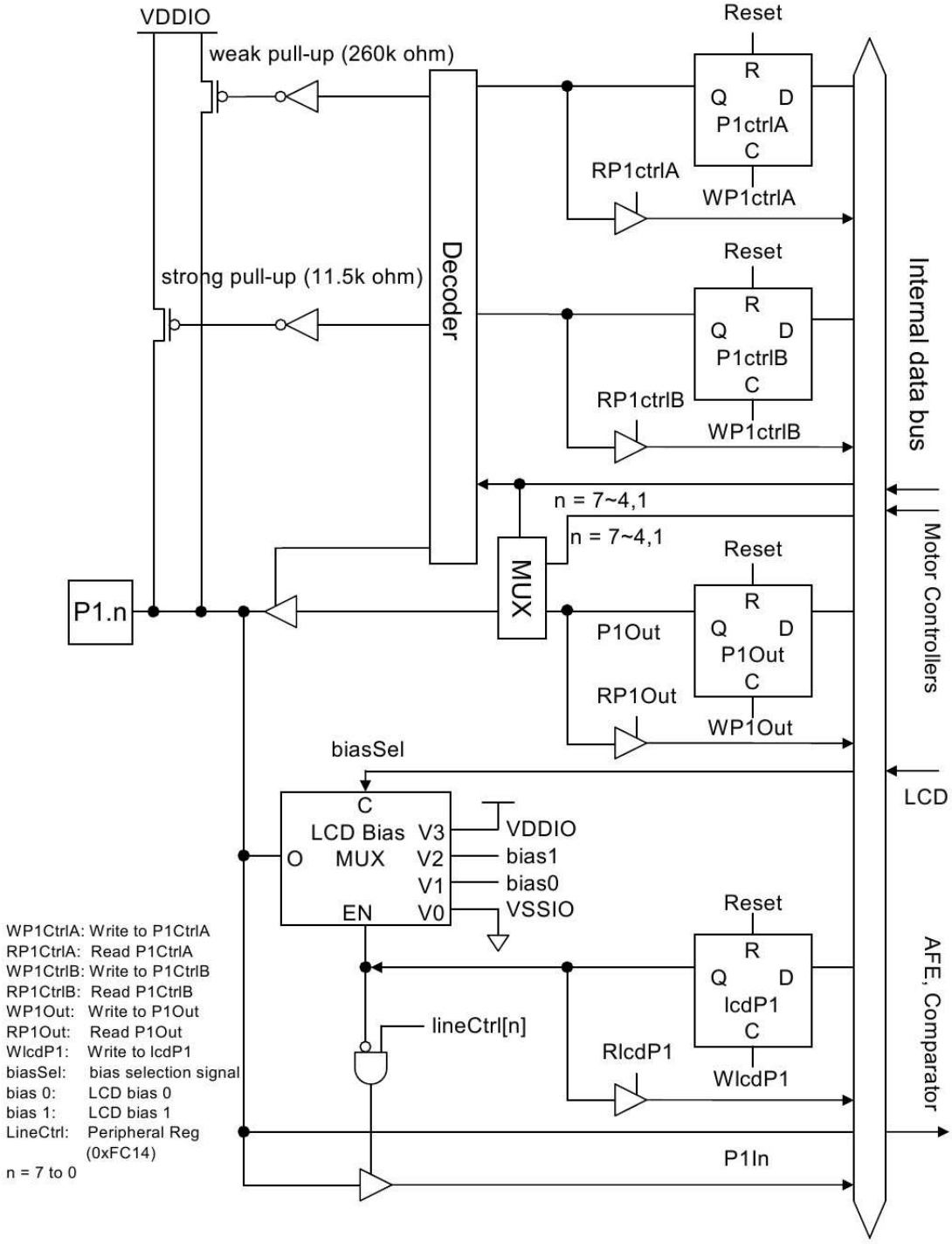
General Purpose I/O Equivalent Circuits



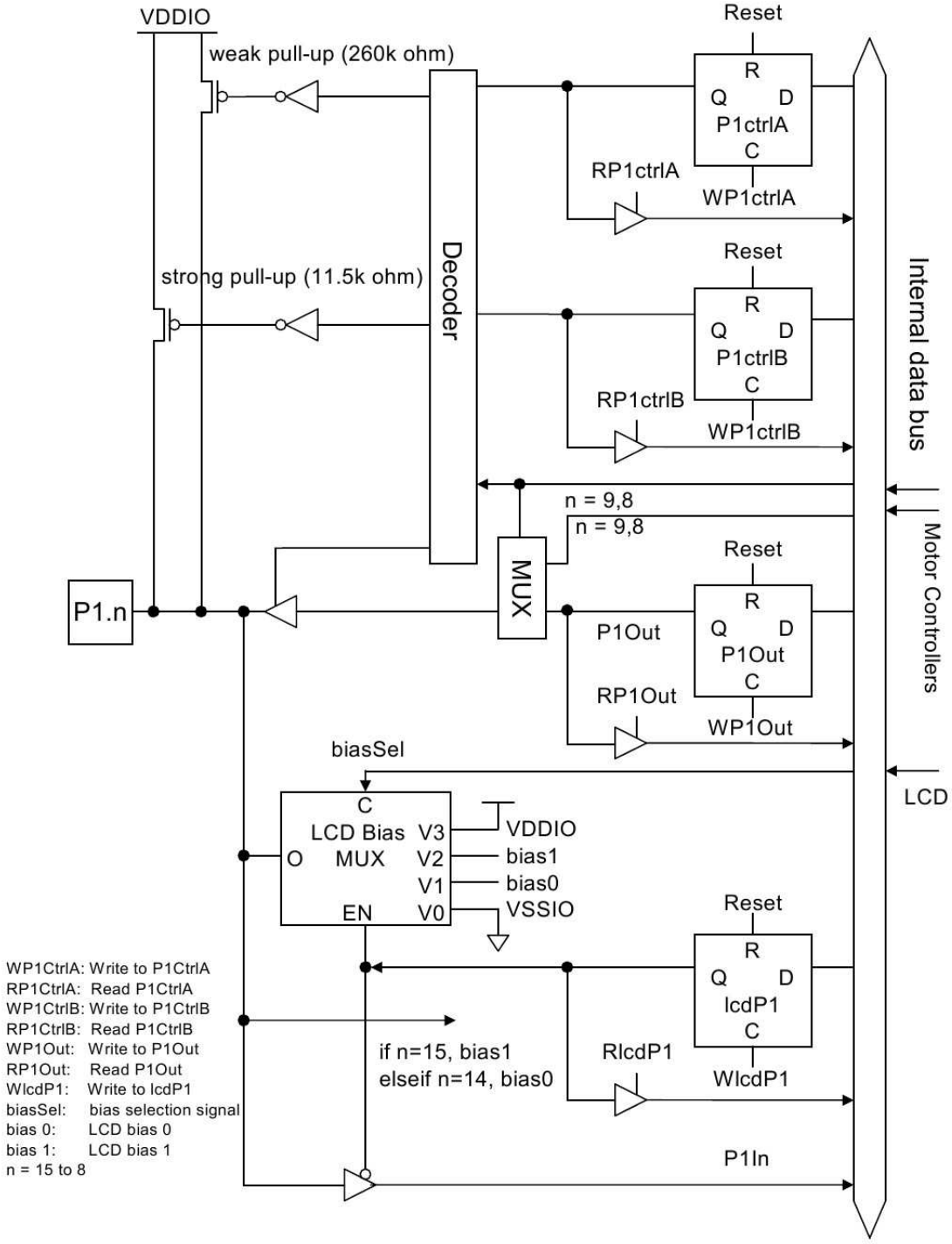
Ports 0 Block Diagram (Pins P0.7 – P0.0)



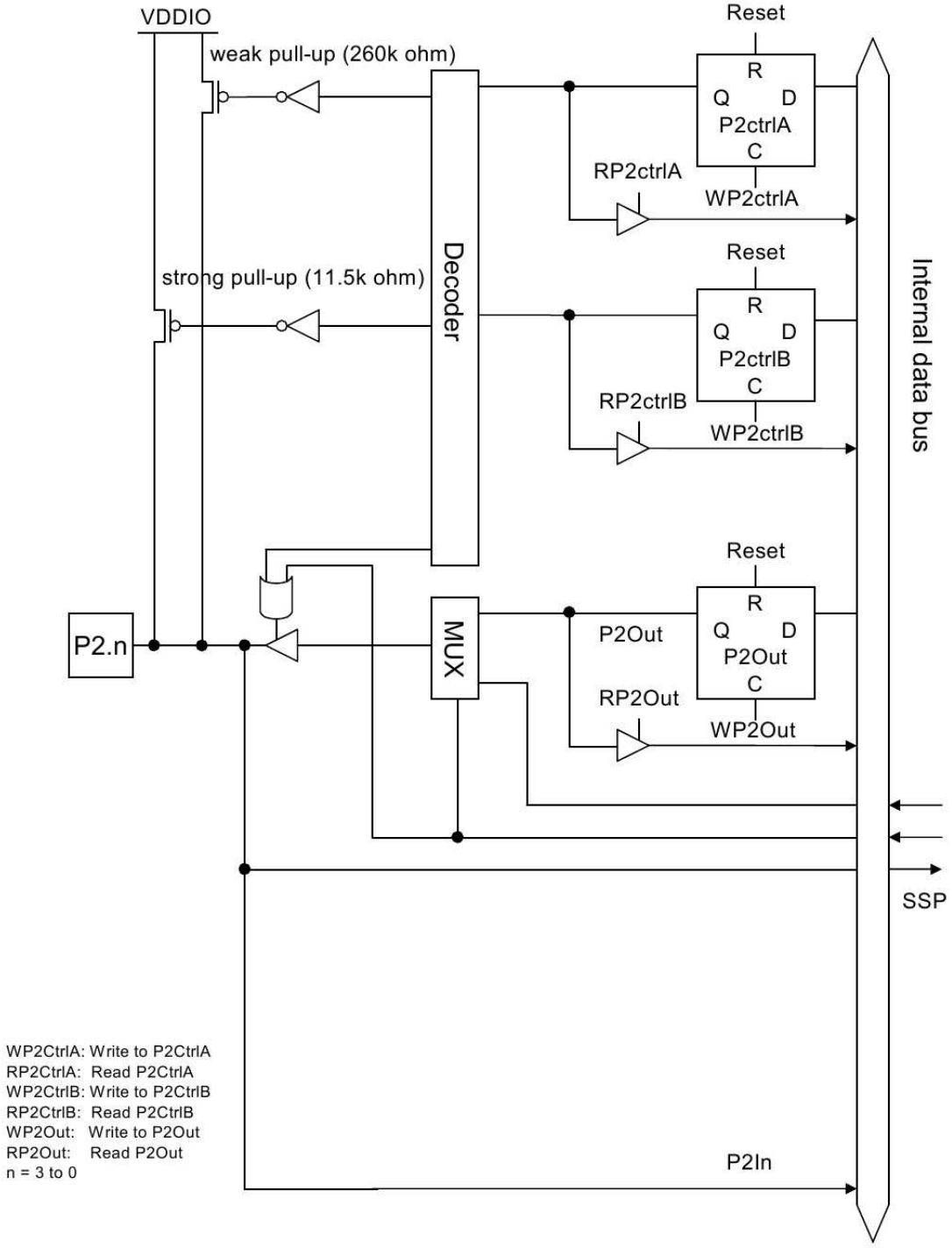
Ports 0 Block Diagram (Pins P0.15 – P0.8)



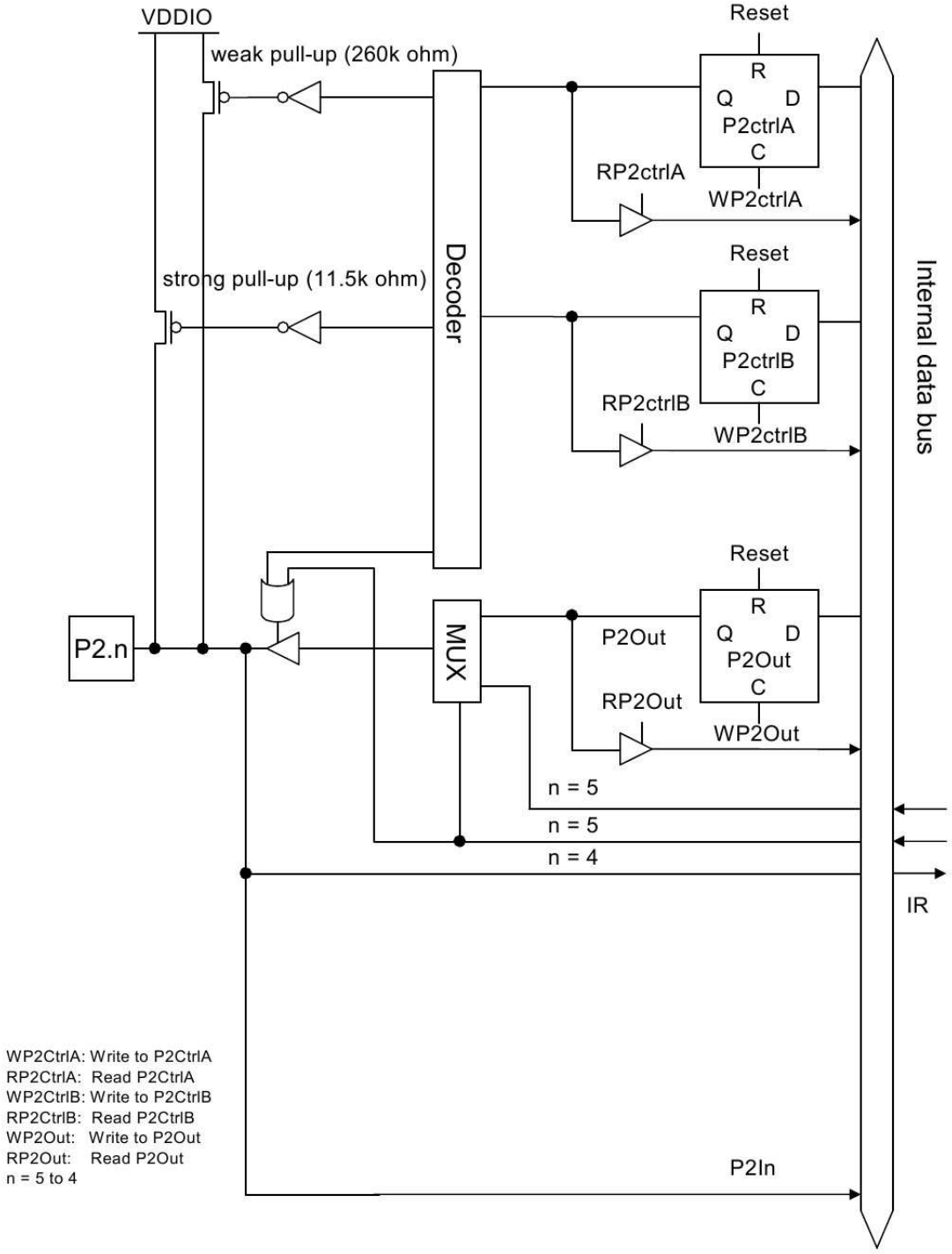
Ports 1 Block Diagram (Pins P1.7 – P1.0)



Ports 1 Block Diagram (Pins P1.15 – P1.8)



Ports 2 Block Diagram (Pins P2.3 – P2.0)



Ports 2 Block Diagram (Pins P2.5, P2.4)

General Purpose Timers

There are four general purpose timers in the NLP-5x. Timer #0 and Timer #1 are integrated with the NLP-5x core. Timer #2 is a peripheral device clocked by one of the two slow speed oscillators (OSC #2 or OSC #3). Timer #3 is a peripheral device clocked by the memory clock, MEMCLK, which is the same rate as the processor clock, CPUCLK, except in turbo mode, in which case it is CPUCLK divided-by-2.

Timers #0 and #1

These two timers are part of the NLP-5x core and are configured via control registers %tc, %timer0, and %timer1. These timers run at the rate of the processor clock (CPUCLK) in non-turbo mode, and at the rate of the processor clock divided-by-2 (CPUCLK/2) in turbo mode. The fields in the %tc register enable each timer, set the prescale value for each timer, and set the timer mode. The lower half of the %tc register sets the enable, mode, and prescale values for timer #0, and the upper byte sets these values for timer #1. The values of the %tc, %timer0, and %timer1 registers are zero at reset.

Timer #0 and Timer #1 have corresponding interrupt requests that are set high when the timers decrement to zero. Timer #0 will set %ireq[5], and timer #1 will set %ireq[6].

%tc register

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
et1	cm1	ps_div1						et0	cm0	ps_div0					

%tc is reset to zero.

et1 1: enable timer #1 operation
0: disable timer #1 operation

cm1 1: timer #1 auto-reload mode. In this mode, when the timer #1 counter decrements to zero, the counter register is reloaded automatically with the initial value on the next CPUCLK (CPUCLK/2 in turbo mode).
0: timer #1 stops counting when it decrements to zero.

ps_div1 This 6 bit value determines how often the timer #1 counter is decremented. If zero, the counter is decremented on every clock. If not zero, the timer is decremented every N+1 input clocks.

et0 1: enable timer #0 operation
0: disable timer #0 operation

cm0 1: timer #0 auto-reload mode. In this mode, when the timer #0 counter decrements to zero, the counter register is reloaded automatically with the initial value on the next CPUCLK (CPUCLK/2 in turbo mode).
0: timer #0 stops counting when it decrements to zero.

ps_div0 This 6 bit value determines how often the timer #0 counter is decremented. If zero, the counter is decremented on every clock. If not zero, the timer is decremented every N+1 input clocks.

%timer0, %timer1 registers

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
divisor															

%timer0, %timer1 are reset to zero.

The timer divisor value, M, and the prescale value, N, determine the timer period, T, as follows:

if N=0 (prescale=0),
T = M+1

else (prescale <> 0),
T = M * (N+1)

Timer #2

Timer2 is clocked by the output of either OSC #2 (slow speed crystal oscillator) or OSC #3 (the on-chip RC oscillator). Bit lsoCtrl[4] is used to select which low speed oscillator is the timer #2 clock source. OSC #2 is nominally 32768 Hz. OSC #3, the RC oscillator can be adjusted to be approximately either 128 KHz or 32 KHz.

Timer #2 is enabled by bit 8 of register timerCtrl (0xFC0A).

The overflow pulse from timer #2 can generate an interrupt request. This interrupt request can be used to wake the chip if it is sleep mode. The overflow status flag is bit 14 of register timerCtrl (0xFC0A). This bit will be set whenever timer #2 overflows. It can be cleared by software.

Timer #2 has a 16-bit count register (t2Value) that counts up. If the counter is about to overflow to zero, it is reloaded with the contents of the t2Reload register. For example, if t2Reload = 0xFF80 (-128), the t2Value register will count:

0xFF80, 0xFF81 … 0xFFFE, 0xFFFF, 0xFF80

or

-128, -127, … -2, -1, -128

Thus the t2Reload register contains the 2's complement of the period. The maximum period is 65536, which corresponds to t2Reload = 0.

Timer 2 Registers

Register Name	Register Address	Read/Write	Function
t2Reload	0xFC08	Read/Write	Timer #2 Counter Reload, 2's complement of the period. This register is reset to zero, which corresponds to a period of 65536 clocks.
t2Value	0xFC09	Read Write	Read current counter contents. A write to this registers forces loading of the counter from t2Reload *on the next rising edge of the timer 2 clock*. The actual data written is ignored.
timerCtrl	0xFC0A	Read/Write	Bit 8 is used to enable timer 2. Writing a 1 to bit 12 clears the timer 2 overflow flag. Bit 14 is the timer 2 overflow flag.

Timer #3

Timer #3 consists of an 16-bit reload value register (t3Reload), an 16-bit up-counter (t3Value), and a 4-bit prescale register (timerCtrl[3:0]). The reload register is readable and writeable by the processor. The counter is readable. If the processor writes to the counter, the data is ignored. Instead, the counter is preset to the contents of reload register.

When the counter overflows from FFFF to 0000, a pulse is generated that sets bit 15 of the timerCtrl register. This overflow flag can be an interrupt request. Bit 15 of timerCtrl can also be clear by software by writing a "1" to timerCtrl[13]. Rather than overflowing to zero, the t3Value register is reloaded with the contents of the t3Reload register. For example, if t3Reload = 0xFF80 (-128), the t3Value register will count:

0xFF80, 0xFF81 … 0xFFFE, 0xFFFF, 0xFF80

or

-128, -127, … -2, -1, -128

Thus the t3Reload register contains the 2's complement of the period. The maximum period is 65536, which corresponds to t3Reload = 0.

Timer #3 runs at the rate of the memory clock, MEMCLK, which has the same rate as the processor clock, CPUCLK, except in turbo mode, in which case it has the rate of CPUCLK divided-by-2. Between MEMCLK and the counter for timer #3 is a prescale function that divides down MEMCLK. 4 bits from timerCtrl[3:0] select the prescale divisor as shown in the following table:

Prescale Value	Divisor	Prescale Value	Divisor
0000	2	1000	512
0001	4	1001	1024
0010	8	1010	2048
0011	16	1011	4096
0100	32	1100	8192
0101	64	1101	16384
0110	128	1110	32768
0111	256	1111	65536

In addition to its timing capability, Timer #3 can also be configured as a counter of external events. In this configuration it uses either the rising or falling edge of a signal applied to I/O pin P0.5. The selected transition is internally synchronized by the clock that is the output of the prescale function.

Timer 3 Registers

Register Name	Register Address	Read/Write	Function
t3Reload	0xFC0B	Read/Write	Timer #3 Counter Reload, 2's complement of the period. This register is reset to zero, which corresponds to a period of 65536 clocks.
t3Value	0xFC0C	Read Write	Read current counter contents. A write to this register forces loading of the counter from t2Reload. The actual data written is ignored.
timerCtrl	0xFC0A	Read/Write	Bits [7:0] are used to configure timer 3. Writing a 1 to bit 13 clears the timer 3 overflow flag. Bit 15 is the timer 3 overflow flag.

timerCtrl - Timer #2, #3 Control Register (0xFC0A)

15	14	13	12	11	10	09	08
t3_ovf	t2_ovf	clr_t3ovf	clr_t2ovf				t2_on
07	06	05	04	03	02	01	00
t3_on	polarity	p05_src	t3_gated	t3_ps3	t3_ps2	t3_ps1	t3_ps0

cleared to zero on reset.

Bit [15] R/O: Timer #3 overflow flag. This bit can be used as an interrupt source.

Bit [14] R/O: Timer #2 overflow flag. This bit can be used as an interrupt source.

Bit [13] W/O: Writing a "1" will clear the timer #3 overflow flag. This bit will be zero when read.

Bit [12] W/O: Writing a "1" will clear the timer #2 overflow flag. This bit will be zero when read.

Bit [8] 0: disable Timer #2 from counting/timing.
1: enable Timer #2.

Bit [7] 0: disable Timer #3 and prescale from counting/timing.
The prescale counter value is reset to zero when this bit is zero.
1: enable Timer #3

Bit [6] In gated timing mode (bits [6:5] = 10) this bit determines the polarity of the P0.4 enable signal:

0: timer gated by P0.4 low
1: timer gated by P0.4 high

In counter mode, (bit 5 = 1) this bit determines the polarity of the P0.5 edge that increments the counter:

0: counter increments on rising edge of P0.5
1: counter increments on falling edge of P0.5

Bit [5] 0: use internal MEMCLK for clock source (timing mode)
1: use external events on I/O pin P0.5 for clock source (counting mode)

Bit [4] 0: normal operation
1: timer is gated by P0.4 according to bit 6.

Bit [3:0] Prescale value. See table on page 72.

The meaning of bits [6:4] is summarized in the following table:

Bit6	Bit5	Bit4	Timer Source	Mode of Operation
x	0	0	MEMCLK	timer
0	0	1	MEMCLK	timer gated by P0.4 LOW
1	0	1	MEMCLK	timer gated by P0.4 HIGH
0	1	x	P0.5	count P0.5 events, rising edge
1	1	x	P0.5	count P0.5 events, falling edge

Audio and Analog Circuitry

Analog Front End

The Analog Front End (AFE) of the NLP-5x consists of a successive-approximation Analog-to-Digital Converter (ADC) along with associated reference voltage generators, a line-level input buffer, and 2 independent Programmable Gain Amplifiers (PGAs).

The AFE also has analog multiplexers that select the current input to the ADC. The 3:1 multiplexer at the input to the ADC selects the source of the audio signal to be converted. This multiplexer is controlled by hardware to automatically switch the input source for consecutive samples. The ADC can be configured to sample from 1, 2, or all 3 sources (i.e., do 1, 2, or 3 conversions) during each sample period.

A block diagram of the AFE is shown on the next page. In that diagram, only one of the two identical PGA blocks is shown. Note that resistor values shown in the diagram are typical, and can vary +/- 20%.

Each PGA consists of a microphone pre-amplifier, an off-chip low pass filter, a buffer, and a zero-crossing detector. The block diagram shows how the PGA blocks can be used as a microphone preamplifier. The pre-amplifier is shown in its non-inverting mode of operation. This op amp can also be used in an inverting configuration, typically for line level inputs. In is recommended that the output of the op amp is low-pass filtered with an external RC network. This filtering reduces the aliasing expected in a successive-approximation ADC. The filtered signal is buffered and becomes an input to the ADC. The zero-crossing detector can be used by hardware in the NLP-5x as a signal when to update the gain level register of the pre-amp. This reduces pops caused by changing the gain level during a recording.

The third channel is called the line level channel. The input to the third channel buffer is selected to be one of pins P1[0] to P1[7] via an 8:1 multiplexer. Because P1[7:0] are directly connected to analog functions inside the NLP-5x, it is required that USBVDD (the power supply for the P1[7:0] pins) be less than or equal to AVDD + 0.3V.

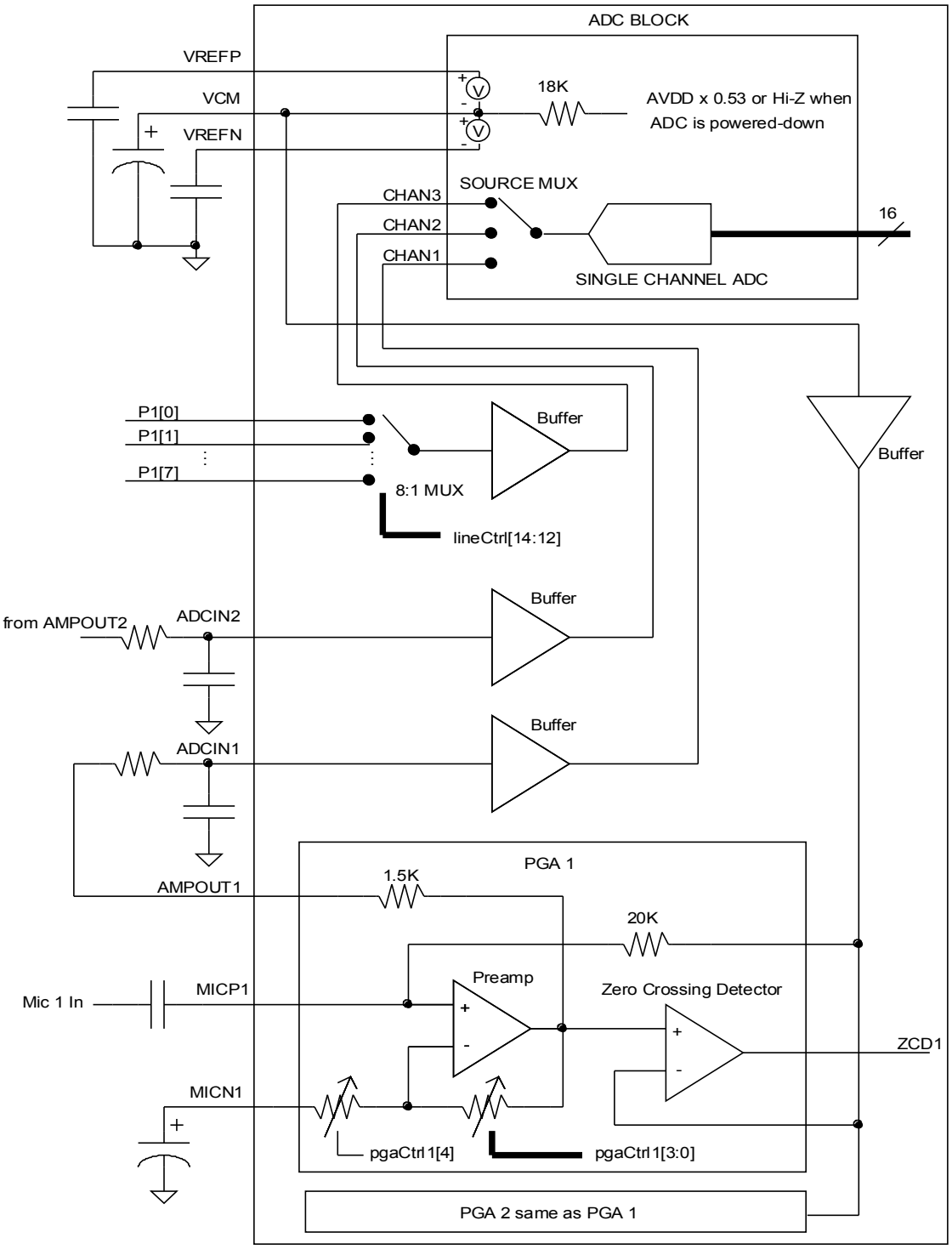
Each major functional block has its own power-down control, and all blocks will be powered down when the ADC is powered down.

Reference Voltages

VCM is generated by a resistor divider between AVDD and AVSS. It is typically 0.53 x AVDD volts. The output impedance of the resistor divider is 18 Kohm +/- 20%. This pin should be bypassed to AVSS with an external capacitor, typically 2.2uF. The VCM generator is disconnected (floating) when the ADC is powered down.

VREFP and VREFN are reference voltages that determine the full-scale input range of the ADC. Each voltage reference is a symmetric voltage offset from VCM. The full-scale input range of the ADC can be selected in software among three choices. Each of VREFP and VREFN should be bypassed to AVSS with 0.01 uF capacitors.

Analog Front End Block Diagram



Programmable Gain Amplifier

There are two independent PGA blocks. Each PGA block is controlled by one of the two PGA control registers, pgaCtrl1 and pgaCtrl2. Bits [4:0] of these registers set the gain of the pre-amplifier. The most significant bit of the gain control determines the value of the inverting input resistor. The 4 least significant bits determine the value of the feedback resistor.

GAIN[4]	Gain Range	MICN1/MICN2 Input Resistor
0	Low Gain Range: AMPOUTx/MICNx = 0 to 22.5 dB in 16 steps	3.6K Typical
1	High Gain Range: AMPOUTx/MICPx = 23 to 43 dB in 16 steps	1.1K Typical

Note that the Low Gain Range is specified relative to input signals at the inverting input (MICN1 or MICN2), while the High Gain Range is specified relative to input signals at the non-inverting input (MICP1 or MICP2) .

For microphone level signals, the high gain range is used. The microphone input is A.C. coupled to either MICP1 or MICP2 with an external capacitor (0.22uF typical). The non-inverting input is biased to VCM with an on-chip 20K typical resistor. The output of the op amp is low-pass filtered with an external RC network to reduce aliasing to be expected by a successive-approximation ADC.

The inverting input to the op amp is usually connected to analog ground with a large capacitor, typically 2.2 uF. This blocks D.C. gain in the amplifier.

The zero-crossing detector is a comparator that detects when the output of the amplifier crosses above VCM. Each PGA can be configured to use the rising edge of the output of this comparator to update the register that provides the gain setting to the PGA with the current contents of the control register, either pgaCtrl1[4:0] or pgaCtrl2[4:0]. In this case the PGA gain level will not change until the signal is near to the idle level, VCM. This reduces pops that might be heard when gain changes on the fly.

If this mode is disabled, the contents of the PGA control register bits [4:0] take effect on the next system clock after the PGA control register is written.

PGA Startup Time

The PGA startup time is partly determined by the time required to charge the VCM capacitor, and partly by the time required to charge the capacitor connected to the inverting input of the op amp (MICN1 and/or MICN2). The former time is determined by the impedance of the VCM on-chip generator and the size of the VCM bypass capacitor. The latter time is determined by the size of the MICN1/2 capacitor and the combination of the on-chip feedback resistor and inverting input source resistor. The combination of the two resistors is smallest when the PGA gain is set to its lowest level, i.e., GAIN[4:0] = 0. In this situation the combination of the feedback resistor and the inverting input resistor is 2 x 3.6K typically.

For fastest startup, software should enable the ADC and the PGA blocks as soon as possible, with the PGA configured with the minimum gain. Once the capacitors are charged, the gain can be changed to the desired level.

pgaCtrl1 – Programmable Gain Amplifier #1 Control Register (0xFC12)

15	14	13	12	11	10	09	08
cmpout1				lpwr1	cmpon1	paon1	bufon1
07	06	05	04	03	02	01	00
zcmode1			gain1[4]	gain1[3]	gain1[2]	gain1[1]	gain1[0]

cleared to zero on reset.

pgaCtrl2 – Programmable Gain Amplifier #2 Control Register (0xFC13)

15	14	13	12	11	10	09	08
cmpout2				lpwr2	cmpon2	paon2	bufon2
07	06	05	04	03	02	01	00
zcmode2			gain2[4]	gain2[3]	gain2[2]	gain2[1]	gain2[0]

cleared to zero on reset.

Bit [15] R/O, zero-crossing detector comparator output state

Bit [11] 0: normal power mode
1: low power mode

Bit [10] 0: zero-crossing detector comparator powered down
1: zero-crossing detector comparator powered up. When the ADC is powered down, this bit is ignored, and the comparator is powered down..

Bit [9] 0: preamp powered down
1: preamp powered up. When the ADC is powered down, this bit is ignored, and the preamp is powered down..

Bit [8] 0: buffer powered down
1: buffer powered up. When the ADC is powered down, this bit is ignored, and the buffer is powered down..

Bit [7] If zcmode is 0 when this register is written, the gain value is copied from the control register to the gain register in the next clock cycle after the write.

If zcmode is 1 when this register is written, the gain value is copied from the control register to the gain register on the next positive zero crossing. This flag will be cleared after the gain value is copied.

Bit [4] 0: line level range
1: mic level range

Bits [3:0] Gain level, 0 to 15, depends on bit 4:
Line level range: 0 to 22.5 dB (based on inverting configuration)
Mic level range: 23 to 43 dB (based on non-inverting configuration).

Line-level Channel

The third input channel to the ADC (Analog-to-Digital Converter) is connected to a buffer that in turn is connected to one of port P1 pins P1[7:0]. Bits [14:12] of register lineCtrl control the 8:1 analog multiplexer.

The input voltage range for channel 3 the ADC is centered around VCM, and spans from reference voltages VREFN to VREFP.

If the third ADC channel is not used, the buffer should be powered down by clearing bit 9 of lineCtrl.

Because P1[7:0] are directly connected to analog functions inside the NLP-5x, it is required that USBVDD (the power supply for the P1[7:0] pins) be less than or equal to AVDD + 0.3V.
.

lineCtrl – Line Level Input Channel Control Register (0xFC14)

15	14	13	12	11	10	09	08
	c3src2	c3src1	c3src0	lpwr3		bufon3	
07	06	05	04	03	02	01	00
p1.7_dig	p1.6_dig	p1.5_dig	p1.4_dig	p1.3_dig	p1.2_dig	p1.1_dig	p1.0_dig

cleared to zero on reset.

Bits [14:12] Channel 3 source:
0 0 0 P1[0]
0 0 1 P1[1]
.
.
1 1 1 P1[7]

Bit [11] Channel 3 buffer power setting:
0: normal power mode
1: low power mode

Bit [9] Channel 3 buffer enable:
0: channel 3 buffer powered down
1: channel 3 buffer powered up. When the ADC is powered down, this bit is ignored, and the buffer is powered down..

Bits [7:0] Used to enable the digital input buffers of pins P1[7:0]. The input buffers of P1[7:0] can be disabled to save power when a pin is used for an analog input signal (ADC channel 3 input, and/or comparator input). *By default after reset the digital input buffers are disabled.*

Note: any of P1[7:0] can be configured as an output even if the input buffer is disabled. A weak or strong pullup device can be enabled even if the input buffer is disabled.

A/D Converter (ADC)

The ADC is a successive-approximation converter that can multiplex up to 3 input channels. The sample rate can vary from 8 KHz to 96 KHz for each channel, with 48 KHz being the typical sample rate. At the typical sample rate and in the middle voltage range (2.4-3.3V) the ADC provides 16-bits of resolution, 70 dB or better SNR from 20 to 8 KHz, and harmonic distortion less than 0.1%.

The clock to the ADC is normally set to be 4 MHz. It is generated by dividing down PLLCLK by a divisor specified in bits [5:1] of register adcCtrl1.

The ADC can be configured to suit the AVDD power supply voltage, in one of three ranges, selected by bits [3:2] of register adcCtrl2:

- Regulated 1.8V +/- 10%
- Typical: 2 batteries, unregulated, 2.4V to 3.3V. The ADC will continue to operate with somewhat reduced dynamic range, noise and/or THD performance as the battery voltage drops below 2.4V.
- Regulated 3.3V +/- 5%.

The first two input channels of the ADC are connected to on-chip buffers that are driven by input pins ADCIN1 and ADCIN2. Typically, these are connected externally to the outputs of the programmable gain amplifiers (PGAs). The third channel of the ADC is connected to an on-chip buffer that is driven by one of pins P1.0-P1.7. The current input pin for the third channel is selected by register lineCtrl, pg 68.

ADC Control and Operation

Analog-to-Digital conversions are configured using the ADC Control Registers **adcCtrl1**, **adcCtrl2**, and **acdSR**, described below. ADC results are stored in a FIFO arranged as 4 ranks, each holding 1, 2, or 3 conversion results. The FIFO is accessed via registers **adcFifo1**, **adcFifo2**, and **adcFifo3**. The FIFO has a status register, **adcStatus.**

The ADC clock is typically 4 MHz, and is derived from PLLCLK. Bits [5:0] of adcCtrl1 specifies the divisor used to generate the ADC clock.

The ADC sample conversion process starts with a sample pulse being sent to the ADC. This pulse samples the voltage on one of the three input channels. The time between successive sample pulses determines the sample rate of the ADC. If two channels are being converted, the sample rate of each channel is one-half of the ADC sample rate. If three channels are being converted the sample rate of each channel is one-third of the ADC sample rate. The time between successive conversions is specified by a programmable number of ADC clocks, plus a programmable number of PLLCLK clocks. Register adcSR, the ADC Sample Rate Register is used to provide these values.

One, two, or three back-to-back ADC conversions can be grouped together in a “frame”. Each conversion period within a frame is called a “slot”. For example, if conversion of all three source channels is required, a frame consists of a three slots: conversion of channel 1, followed by a conversion of channel 2, followed by a conversion of channel 3 (though the channel order is arbitrary). After three slots are completed, the ADC result for all three channels is written to the FIFO at the same moment.

Register adcCtrl1, the ADC Control Register #1, defines the number of slots per frame, and assigns channel numbers to each slot. Note: all three slots of a frame can be assigned to the same channel to maximize the use of the FIFO when only a single channel is being converted.

adcCtrl1 – ADC Control Register #1 (0xFC20)

15	14	13	12	11	10	09	08
s3ch1	s3ch0	s2ch1	s2ch0	s1ch1	s1ch0	nslots1	nslots0
07	06	05	04	03	02	01	00
		adcClk4	adcClk3	adcClk2	adcClk1	adcClk0	adcclk_on

cleared to zero on reset.

Bits [15:14] Timeslot #3 channel source:
0 0 channel 1 (adcin1)
0 1 channel 2 (adcin2)
1 0 channel 3 (selectable P1.0-P1.7)
1 1 ---

Bits [13:12] Timeslot #2 channel source:
0 0 channel 1
0 1 channel 2
1 0 channel 3
1 1 ---

Bits [11:10] Timeslot #1 channel source:
0 0 channel 1
0 1 channel 2
1 0 channel 3
1 1 ---

Bits [9:8] Number of timeslots per sample frame:
0 0 1 slot
0 1 2 slots
1 0 3 slots

Bits[7:6] Reserved. These bits should be written with zeros.

Bits [5:1] ADC Clock Generator Divisor. Nomally, the ADC is clocked at 4 MHz, which is generated from the PLL output. For example, if the PLL output is 72 MHz, a divisor of 18 would produce a 4 MHz clock to the ADC.

1-31 divide by 1 to 31
0 divide by 32

Bit [0] Enable ADC Clock Generator
0 ADC clock generator and associated FIFO is disabled and held in a reset state.
1 ADC clock generator is enabled.
Note: the ADC analog block has a separate enable: bit 0 of ADC Control Register #2.

adcCtrl2 – ADC Control Register #2 (0xFC21)

15	14	13	12	11	10	09	08
						fdint1	fdint0
07	06	05	04	03	02	01	00
div2	div1	div0		inVolt1	inVolt0		adc_on

cleared to zero on reset.

Bits[9:8] FIFO depth to produce an interrupt request
0 0 Interrupt generated after one frame in FIFO
0 1 Interrupt generated after two frames in FIFO
1 0 Interrupt generated after three frames in FIFO
1 1 Interrupt generated after four frames in FIFO

Bit [7:5]: ADC internal clock divider
0 0 0 divide by 1
0 0 1 divide by 1.5
0 1 0 divide by 2
0 1 1 divide by 3
1 0 0 divide by 4.5
1 0 1 divide by 6
1 1 0 divide by 9
1 1 1 divide by 12

Bits [3:2]: ADC Voltage input select.
0 0 1.8V regulated
0 1 >= 2.4V unregulated
1 0 >= 3.3V regulated
1 1 (unused)

Bit [0]: Enable ADC circuit and ADC buffers.
1 ADC on
0 ADC powered down
Preamps must be enabled separately

adcCtrl3 – ADC Control Register #3 (0xFC23)

15	14	13	12	11	10	09	08
07	06	05	04	03	02	01	00
			phase	dlysel3	dlysel2	dlysel1	dlysel0

cleared to zero on reset.

This register can be used to select a delay for the ADC clock relative to PLLCLK output and derivative clocks. If dlysel[3:0] is 00xx, minimal delay is added to the ADC clock. Otherwise, phase = 1 will delay the ADC clock by one-half of the PLL clock period, and dlysel[3:0] specifies additional delay:

ADC clock delay (nsec) ~= 0.4 + 0.65 * (dlysel[3:0]-4)

adcSR – ADC Sample Rate Register (0xFC22)

15	14	13	12	11	10	09	08
smpWidth1	smpWidth0	sr8	sr7	sr6	sr5	sr4	sr3
07	06	05	04	03	02	01	00
sr2	sr1	sr0	adjust4	adjust3	adjust2	adjust1	adjust0

cleared to zero on reset.

Bits [15:14] Specify the width of the SAMPLE pulse to the ADC, in units of the ADC clock. The width should be selected to be as wide as possible based on the sample rate and number of slots.

0 0 2 ADC clocks
0 1 4 ADC clocks
1 0 8 ADC clocks
1 1 16 ADC clocks

Bits [13:5] ADC Sample Rate Period. Number of ADC Clocks per channel sample conversion, including SAMPLE pulse width.

Bits [4:0] ADC Sample Rate Adjust. Number of PLL clocks to add to stretch the first ADC clock of each channel sample conversion to tune the sample rate. This field is ignored when the value of the ADC Clock Generator Divisor is 1 (i.e., adcCtrl[5:1] = 00001).

Depending on the system clock, some sample rates can be achieved exactly and others can be approximated with acceptable accuracy. The following table shows how various sample rates can be generated based on the number of ADC channels and the system clock rate.

Table of Typical ADC Control Settings

PLL VCO/2 Output Rate	Desired Channel Rate	# Channels	ADC Rate	ADC SR Period	ADC SR Adjust	Actual Channel Rate	Error
80 MHz	48 KHz	3	144 KHz	27	16/20	47.962	-.08%
80 MHz	48 KHz	2	96 KHz	41	13/20	48.019	+.04%
80 MHz	48 KHz	1	48 KHz	83	7/20	47.990	-.02%
80 MHz	32 KHz	3	96 KHz	41	13/20	32.013	+.04%
80 MHz	32 KHz	2	64 KHz	62	10/20	32.000	---
80 MHz	32 KHz	1	32 KHz	125	0/20	32.000	---
40 MHz	48 KHz	3	144 KHz	27	8/10	47.962	-.08%
40 MHz	48 KHz	2	96 KHz	41	7/10	47.962	-.08%
40 MHz	48 KHz	1	48 KHz	83	3/10	48.019	+.04%
40 MHz	32 KHz	3	96 KHz	41	7/10	31.974	-.08%
40 MHz	32 KHz	2	64 KHz	62	5/10	32.000	---
40 MHz	32 KHz	1	32 KHz	125	0/10	32.000	---
36 MHz	48 KHz	3	144 KHz	27	7/9	48.000	---
36 MHz	48 KHz	2	96 KHz	41	6/9	48.000	---
36 MHz	48 KHz	1	48 KHz	83	3/9	48.000	---
36 MHz	32 KHz	3	96 KHz	41	6/9	32.000	---
36 MHz	32 KHz	2	64 KHz	62	5/9	32.028	-.09%
36 MHz	32 KHz	1	32 KHz	125	0/9	32.000	---
4 MHz	96 KHz[*1]	3	288 KHz	14	n/a	95.238	-.79%
4 MHz	48 KHz	3	144 KHz	28	n/a	47.619	-.79%
4 MHz	48 KHz	2	96 KHz	42	n/a	47.619	-.79%
4 MHz	48 KHz	1	48 KHz	83	n/a	48.193	+.40%
4 MHz	32 KHz	3	96 KHz	42	n/a	31.746	-.79%
4 MHz	32 KHz	2	64 KHz	63	n/a	31.746	-.79%
4 MHz	32 KHz	1	32 KHz	125	n/a	32.000	---

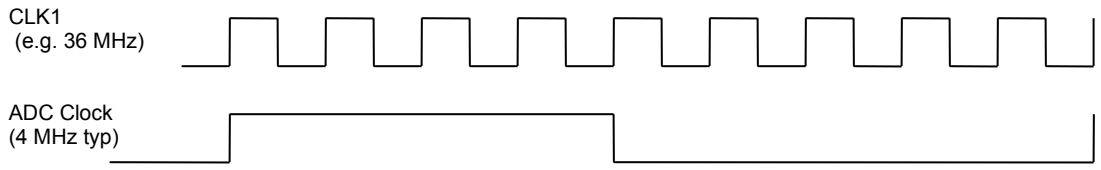
*1: 96 KHz x 3 is the worst case sample rate. It must complete in 14 ADC clocks.

Example:

If the system clock is 40 MHz, and the ADC clock divisor (0xFC20[5:1]) is set to 10, then the ADC Clock is 4 MHz. If we want to convert 3 channels at 48 KHz, we must program the ADC for a sample rate of 3 x 48 = 144 KHz. This corresponds to 4 MHz/144 KHz = 27.778 ADC clocks. We approximate this by setting the ADC conversion period to 27 ADC clocks, plus an extra number of system clocks. 8 system clocks is 8/10 = 0.8 ADC clocks, so the total sample conversion period is 27.8 ADC clocks, or 4 MHz/27.8 = 143.885 KHz. The sample rate per channel is 143.885/3 = 47.962 KHz. The error is -0.08%.

ADC Clock Timing Example

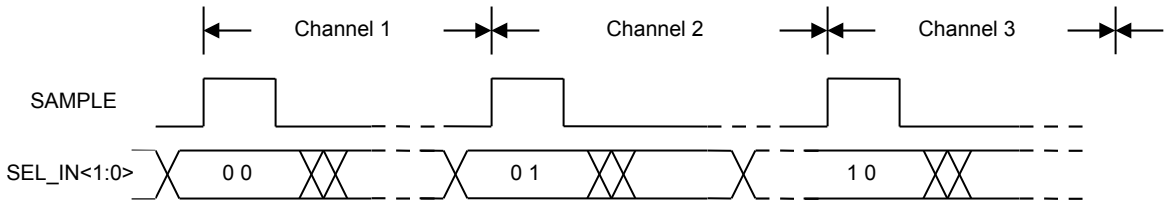
The following diagram shows the ADC Clock generated from the system clock which is 36 MHz in this example. The ADC Clock divisor is 9. Note that since the divisor is odd, the duty cycle of the ADC clock is not 50/50.



Sample Frame Example – 3 Slots x 48 KHz, 36 MHz System Clock

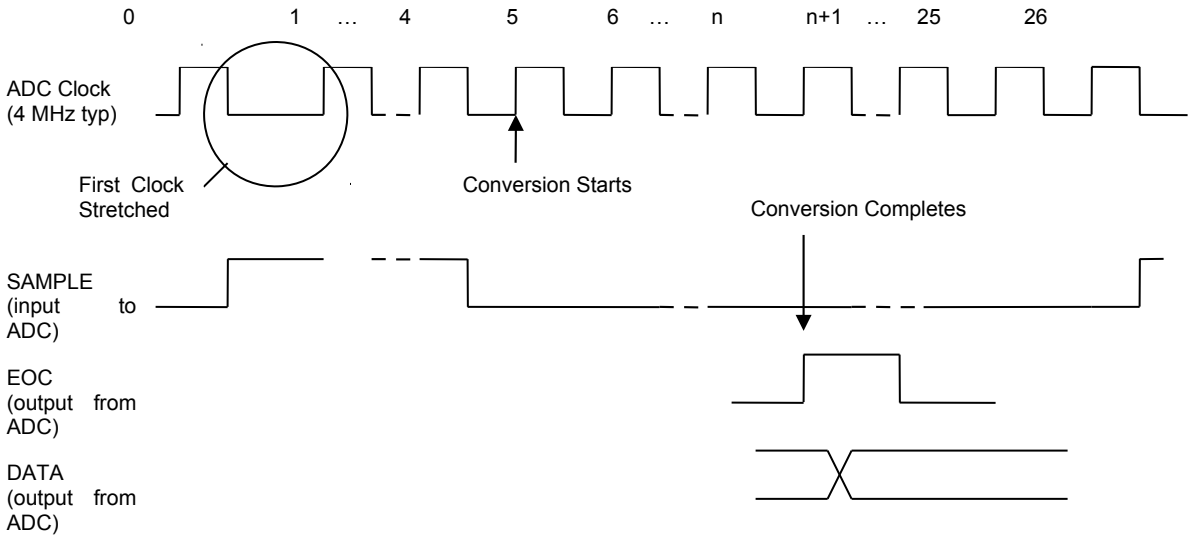
In this example we want to convert 3 slots at 48 KHz. The ADC must convert 48 x 3 = 144K samples per second. The system clock rate is 36 MHz. The ADC clock is 4 MHz. The slots are assigned to channels 1, 2 and 3 respectively.

The SAMPLE pulse signals the period when the ADC samples its input. The SEL_IN<1:0> bus determines which of the 3 ADC channels is sampled in a given slot. The SAMPLE pulse is an integer number of ADC clock periods in width, determined by bits 15 and 14 of register 0xFC22.



The timing for each slot sample conversion is identical, and is shown below. In this example the SAMPLE pulse is programmed to be 4 ADC clock periods wide, plus 7 additional system clocks. Note that the SAMPLE signal is generated by the falling edge of the ADC clock.

Conversion starts on the rising edge of clock period following the falling edge of SAMPLE. Conversion requires at least 12 clock periods to complete, but must complete within the overall 27 clock period.



When conversion of one sample is complete for all enabled slots, the converted values for all three slots are transferred simultaneously into the next available rank in the ADC FIFO. Zeros are transferred for ADC slots that are not enabled. If the number of FIFO ranks holding data is larger than the number specified by bits [9:8] of ADC Control Register #2, an ADC interrupt request is generated. The ADC interrupt request is connected to external interrupt #4 of the core. If the FIFO is full *prior* to being written, the new data is discarded. Reading an ADC slot if the FIFO is empty will return zero.

The circuitry is designed to guarantee that valid data is returned when reading any of the FIFO registers at the same moment as they are being loaded.

The read index of the FIFO is updated after reading register ADC Slot #1. It is not affected by reading registers ADC Slot #2 or ADC Slot #3. The FIFO is reset to be empty if ADCCLK_ON is clear (note: not ADC_ON), or by writing to ADC Slot #1.

adcFifo3 – ADC FIFO Slot #3 Register (0xFC24)

15	14	13	12	11	10	09	08
adc3_15	…						
07	06	05	04	03	02	01	00
							adc3_00

This register contains the signed 16-bit result of the oldest ADC conversion of slot 3 in the FIFO. It is a read-only register. This register returns 0 if the FIFO is empty or if slot 3 is disabled.

adcFifo2 – ADC FIFO Slot #2 Register (0xFC25)

15	14	13	12	11	10	09	08
adc2_15	…						
07	06	05	04	03	02	01	00
							adc2_00

This register contains the signed 16-bit result of the oldest ADC conversion of slot 2 in the FIFO. It is a read-only register. This register returns 0 if the FIFO is empty or if ADC slot 2 is disabled..

adcFifo1 – ADC FIFO Slot #1 Register (0xFC26)

15	14	13	12	11	10	09	08
adc1_15	…						
07	06	05	04	03	02	01	00
							adc1_00

This register contains the signed 16-bit result of the oldest ADC conversion of slot 1 in the FIFO. Writing any value to this register will reset the FIFO block to be empty for all channels. Writing to this register will also clear the IRQ and Overrun flags in the status register. After reading this register, the read index for the FIFO is incremented. This register returns 0 if the FIFO is empty.

adcStatus – ADC Status Register (0xFC27)

15	14	13	12	11	10	09	08
afifo_ne	afifo_irq	adc_overrun					
07	06	05	04	03	02	01	00
					fcount2	fcount1	fcount0

bit [15] FIFO not empty.
bit [14] IRQ is set whenever the number of samples in the FIFO exceeds the FIFO[1:0] value in ADC Control Register. Cleared by writing any value to this register.
bit [13] Error flag that is set if an ADC sample is missed because the FIFO was full. Cleared by writing any value to this register.
bit [2:0] (R/O) Number of samples in FIFO.

Pulse Width Modulator (PWM)

The PWM consists of circuitry to generate periodic pulses that have duty cycles proportional to the amplitude of digital audio data. There are two low impedance outputs of the PWM. For positive audio samples, the output PWM0 will drive high while the output PWM1 will be held low. For negative audio samples, the output PWM1 will drive high while the output PWM0 will be held low. For zero value audio samples, or if the PWM is disabled, both PWM0 and PWM1 will be low.

The PWM0 and PWM1 pins can be directly connected to a speaker with an impedance of 8 ohms or greater. The PWM drivers have their own power and ground pins. The power supply for the PWM (PWMVDD) can be as large as 3.6V.

The PWM is controlled by two control registers, a 6-level FIFO, and a status register.

Timing

The PWM timing is controlled by the PWM Control Register, pwmCtrl, and the PWM Frame Width Register, pwmFrame. 3 prescaler bits in pwmCtrl determine how the PLLCLK is divided prior to being used as the PWM clock. 11 bits in pwmFrame determine the number of PWM clocks in each PWM frame. The maximum frame width is 2047 PWM clocks. Each audio sample taken from the PWM FIFO can be used for 1, 2, 3 or 4 consecutive frames. 2 bits in pwmFrame determine how many frames are generated for each sample. Thus these 2 bits specify the ratio between the PWM frame rate and the audio sample rate.

A single PWM frame consists of a pulse to one of the two PWM outputs based on the sign of the current sample. The width of the pulse depends on the magnitude of the sample after a programmable right shift. A PWM frame will try to be symmetric as much as possible about the mid-point, so the output pulse will initially be low, then high for a time based on the magnitude of the sample, then low again.

If the magnitude of the audio sample, after the programmable right shift is applied, is equal to or larger than the frame width, the pulse will be high for the entire frame (100% duty cycle).

The start of the first frame can be optionally synchronized to the ADC end-of-frame signal, or the DAC start-of-frame signal. The PWM will be free-running thereafter.

PWM FIFO

A 6-level FIFO reduces the interrupt service overhead for transferring audio data to the PWM. Bits in the pwmCtrl register are used to specify the required number of empty locations in the FIFO prior to the signaling of an interrupt request.

pwmCtrl – PWM Control Register 0xFC28

15	14	13	12	11	10	09	08
	pwmfid2	pwmfid1	pwmfid0		pwmsh2	pwmsh1	pwmsh0
07	06	05	04	03	02	01	00
	pwmps2	pwmps1	pwmps0		syncdac	syncadc	pwm_en

cleared to zero on reset.

Bits [14:12] FIFO interrupt depth level
- 0 Interrupt happens whenever there is 1 or more empty locations in the FIFO
- 1 Interrupt happens whenever there are 2 or more empty locations in the FIFO
- 2 Interrupt happens whenever there are 3 or more empty locations in the FIFO
- 3 Interrupt happens whenever there are 4 or more empty locations in the FIFO
- 4 Interrupt happens whenever there are 5 or more empty locations in the FIFO
- 5 Interrupt happens whenever the FIFO is empty

Bits [10:8] PWM data shift right count, 0 to 7

Bits [6:4] Clock Prescale
- 0 0 0 Source Clock / 1
- 0 0 1 Source Clock / 2
- 0 1 0 Source Clock / 3
- 0 1 1 Source Clock / 4
- 1 0 0 Source Clock / 5
- 1 0 1 Source Clock / 6
- 1 1 0 Source Clock / 7
- 1 1 1 Source Clock / 8

Bit [2] Set this bit *before* setting PWM enable in order to synchronize the *first* PWM frame with the next DAC start-of-frame pulse.

Bit [1] Set this bit *before* setting PWM enable in order to synchronize the *first* PWM frame with the next ADC end-of-frame pulse.

Bit [0] PWM Enable
- 0 PWM disable, and reset
- 1 PWM enabled

pwmFrame – PWM Frame Width Register (0xFC29)

15	14	13	12	11	10	09	08
		pwmrpt1	pwmrpt0		pwmfw10	pwmfw9	pwmfw8
07	06	05	04	03	02	01	00
pwmfw7	pwmfw6	pwmfw5	pwmfw4	pwmfw3	pwmfw2	pwmfw1	pwmfw0

cleared to zero on reset.

Bits [13:12] Repeat Count
0 0 Each FIFO output is used once
0 1 Each FIFO output is used twice
1 0 Each FIFO output is used three times
1 1 Each FIFO output is used four times

Bits [10:0] PWM Frame Size
The PWM frame counter counts from 0 to N-1, then repeats. The maximum frame size is 2047 PWM clocks.

Note: it is recommended that the PWM be disabled (i.e., bit 0 of pwmCtrl set to 0) when the contents of this register are changed.

pwmFifo – PWM FIFO Register (0xFC2A)

15	14	13	12	11	10	09	08
pwmD15	pwmD14	pwmD13	pwmD12	pwmD11	pwmD10	pwmD09	pwmD08
07	06	05	04	03	02	01	00
pwmD07	pwmD06	pwmD05	pwmD04	pwmD03	pwmD02	pwmD01	pwmD00

This register is used to write data to the PWM FIFO. If the FIFO is full prior to a write operation, the data will be written and the oldest value in the FIFO will be discarded.

Reading this register will reset the FIFO block. A FIFO reset will set the FIFO to be empty. The data returned from the read will reflect the state of the PWM status register after the FIFO reset, with the IRQ and Underrun flags cleared.
.

pwmStatus – PWM Status Register (0xFC2B)

15	14	13	12	11	10	09	08
pfifo_nf	pfifo_irq	pwm_underrun					
07	06	05	04	03	02	01	00
					fcount2	fcount1	fcount0

bit [15] FIFO not full.
bit [14] IRQ is set whenever the number of empty locations in the FIFO exceeds the pwmfid[2:0] field from bits [14:12] of the PWM Control Register. Cleared by writing any value to this register.
bit [13] Error flag that is set if a PWM sample is missed because the FIFO was empty. Cleared by writing any value to this register.
bit [2:0] (R/O) Number of empty locations in the FIFO, 0 to 6. When the FIFO is empty, fcount[2:0] will be 6.

This register will return 0x8006 when the PWM is reset (pwmCtrl[0] = 0).

Comparator Unit

The Comparator Unit consists of two independent analog comparators designated “A” and “B”, two programmable 64-level voltage references, input selection circuitry to select comparator inputs from P1[7:0], and two control registers.

A comparator has two analog inputs, designated “+” and “-“, and one digital output. When the analog voltage on the “+” input is greater than the analog voltage on the “-“ input, the digital output is a high level; when the analog voltage on the “+” input is less than the analog voltage on the “-“ input, the digital output is a low level.

Each comparator can be separately enabled or disabled. When a comparator is disabled, its output is pulled low and the comparator power is turned off. If both comparators are disabled, the reference generator resistor string is disconnected from ground so it does not draw current.

The non-inverting input of each comparator can be assigned to any of ports P1[7:0] under software control. The inverting input of each comparator can be assigned to any of ports P1[7:0], or, to an internal reference generator. The reference generator is a resistor string connected between analog ground and analog Vdd. There are 64 evenly spaced taps centered around VDD/2 to span the input range of the comparator. Note that each comparator has an independent reference selector which shares the same resistor string.

The outputs of the comparators can be read by software, and optionally connected to the output drivers of P1.8 and/or P1.9 for comparators A and B respectively. The rising edge of the comparator outputs can also be used as interrupt sources.

The voltage select value, 0xFC10[13:8] (0xFC11[13:8]) selects one of 64 outputs of an analog mux connected to 64 equally spaced taps on a resistive divider. Each tap is spaced from the adjacent one by a resistor of value, R. The AVSS end of the ladder has an additional resistor of value R/2, and the AVDD end of the ladder has an additional resistor of value R/2. The total resistance (ignoring the power switch resistance) is thus R/2+R/2+63*R=64R. The Comparator Reference Voltage then covers the range from AVDD/128 to AVDD * (127/128):

AVDD * (2 * bits[13:8] + 1)/128

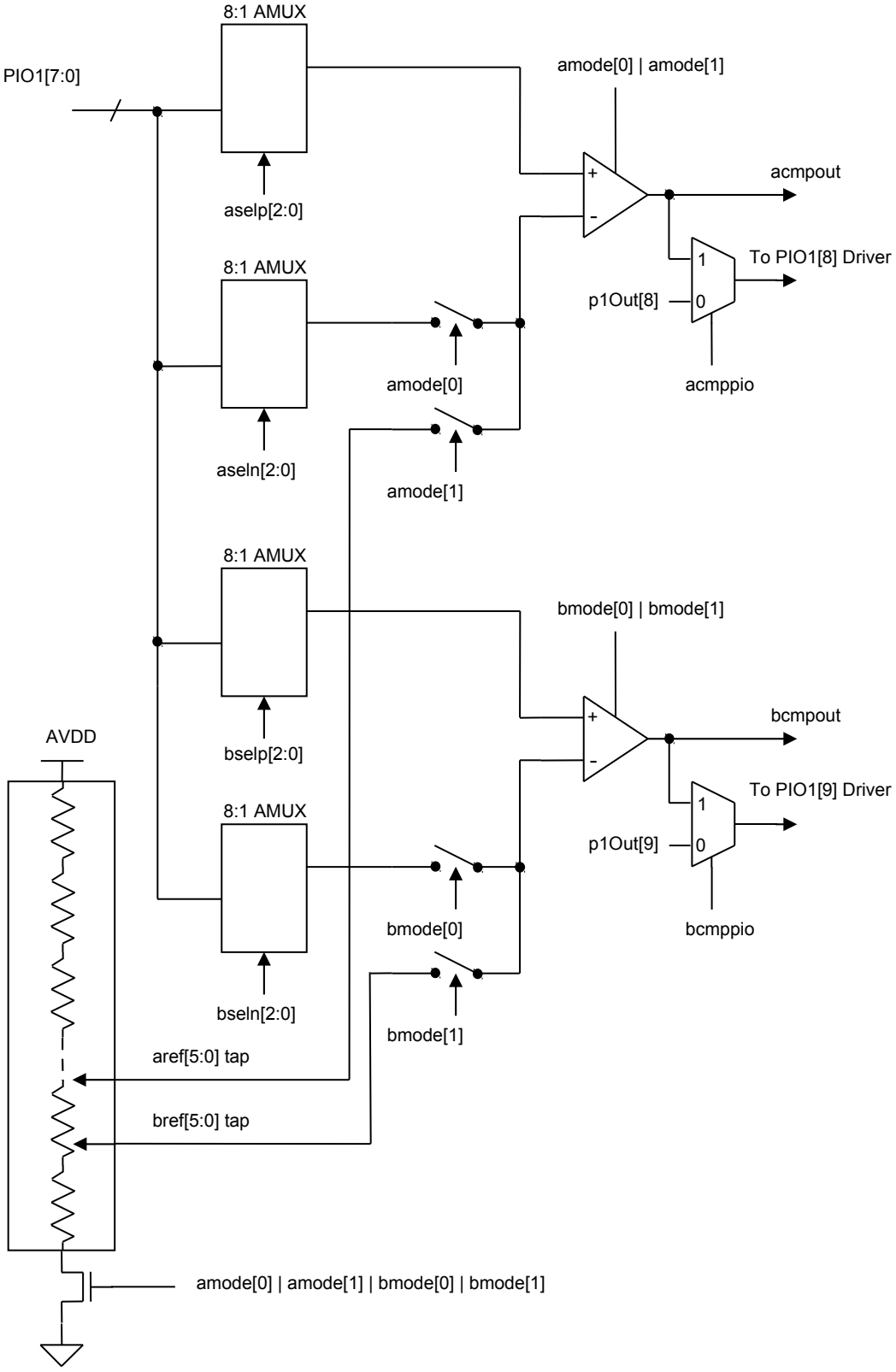
For AVDD of 3.3V, one step corresponds to AVDD/64 = 51.56 mV.

The overall value of the resistor string (64*R) is typically about 50K ohms.

In some configurations the Comparator Control register(s) can be set up once and simply read thereafter. In most configurations it will be necessary to switch the input muxes and/or re-program the appropriate reference voltage repeatedly.

Because P1[7:0] have analog functions, it is required that USBVDD (the power supply for the P1[7:0] pins) be less than or equal to AVDD + 0.3V.

Comparator Block Diagram



The Comparator Control registers configure the comparators and provide access to the digital comparator outputs.

cmpACtrl – Comparator A Control Register (0xFC10)

15	14	13	12	11	10	09	08
acmpout	acmppio	aref5	aref4	aref3	aref2	aref1	aref0
07	06	05	04	03	02	01	00
aselp2	aselp1	aselp0	aseln2	aseln1	aseln0	amode1	amode0

cleared to zero on reset.

Bit [15]*1
- 0 Comparator A+ < A-
- 1 Comparator A+ > A-

Bit [14]*2
- 0 P1.8 output data from register p1Out[8]
- 1 P1.8 output data from Comparator A output

Bits [13:8] Reference Voltage selection (1 of 64 levels) for A- input. No function if bit 1 = 0, i.e., if the A- input is not connected to the reference generator.

Bits [7:5] Select one of eight comparator A+ input pins, P1.0-P1.7

Bits [4:2] Select one of eight comparator A- input pins, P1.0-P1.7 No function if bit 0 = 0, i.e., if the A- input is not connected to PIO.

Bits [1:0] Mode:
- 0 0 Powered down
- 0 1 A- connected to one of P1.0 – P1.7 based on [4:2]
- 1 0 A- connected to reference voltage based on [13:8]
- 1 1 A- connected to *both* PIO and reference voltage

*1: Bit 15 is the comparator output and is read-only by the processor.
*2: P1.8 must be configured as a general purpose output, with LCD option disabled.

cmpBCtrl – Comparator B Control Register (0xFC11)

15	14	13	12	11	10	09	08
bcmpout	bcmppio	bref5	bref4	bref3	bref2	bref1	bref0
07	06	05	04	03	02	01	00
bselp2	bselp1	bselp0	bseln2	bseln1	bseln0	bmode1	bmode0

cleared to zero on reset.

Bit [15][*1] 0 Comparator B+ < B-
1 Comparator B+ > B-

Bit [14][*2] 0 P1.9 output data from register p1Out[9]
1 P1.9 output data from Comparator B output

Bits [13:8] Reference Voltage selection (1 of 64 levels) for B- input.
No function if bit 1 = 0, i.e., if the A- input is not connected to the reference generator.

Bits [7:5] Select one of eight comparator B+ input pins, P1.0-P1.7

Bits [4:2] Select one of eight comparator B- input pins, P1.0-P1.7
No function if bit 0 = 0, i.e., if the B- input is not connected to PIO.

Bits [1:0] Mode:
0 0 Powered down
0 1 B- connected to one of P1.0 – P1.7 based on [4:2]
1 0 B- connected to reference voltage based on [13:8]
1 1 B- connected to *both* PIO and reference voltage

*1: Bit 15 is the comparator output and is read-only by the processor.
*2: P1.9 must be configured as a general purpose output, with LCD option disabled.

Stereo D/A Converter (DAC)

There are two sigma-delta DAC channels that can run up to 48 KHz. There is a 6-level FIFO to make transferring data to the DAC more efficient. The ratio between the sample rate and the DAC clock is fixed at 500. Both channels share the same sample rate.

Each DAC channel has two outputs which behave in a differential fashion. Each DAC output can swing between ground (DACVSS) and the DAC supply voltage (DACVDD). When not muted, the two DAC outputs associated with a given channel will have opposite phase.

Each DAC output has an output resistance of 6K ohms +/- 20%. In MUTE mode, or if the DAC is disabled, all DAC outputs will be high-impedance, i.e., floating.

Each DAC pair should have a capacitor connected between the positive and negative phase output pins as close to the NLP-5x chip as possible, in order to filter out high frequencies that might interfere with the adjacent oscillator pins.

dacCtrl – DAC Control Register (0xFC2C)

15	14	13	12	11	10	09	08
mute	dacfid2	dacfid1	dacfid0	chan2_en	chan1_en		source_clk
07	06	05	04	03	02	01	00
dacClk5	dacClk4	dacClk3	dacClk2	dacClk1	dacClk0	syncadc	dac_en

cleared to zero on reset.

Bit [15] Mute
1: DAC outputs are muted (pulled low with Hi-Z devices).
0: DAC outputs are active (if bit 0 is set).
Normally the DAC outputs are muted whenever the DAC is not enabled. This bit mutes the outputs regardless of the DAC enable state.

Bit [14:12] FIFO interrupt depth level
0 Interrupt happens whenever there is 1 or more empty locations in the FIFO
1 Interrupt happens whenever there are 2 or more empty locations in the FIFO
2 Interrupt happens whenever there are 3 or more empty locations in the FIFO
3 Interrupt happens whenever there are 4 or more empty locations in the FIFO
4 Interrupt happens whenever there are 5 or more empty locations in the FIFO
5 Interrupt happens whenever the FIFO is empty

Bit [11] Channel #2 Enable.
If this bit is clear, the FIFO output for this channel is ignored and replaced with zero.

Bit [10] Channel #1 Enable.
If this bit is clear, the FIFO output for this channel is ignored and replaced with zero.

Bit [8] 0: use PLL divide-by-two output for DAC source clock
1: use PLL divide-by-three output for DAC source clock

Bits [7:2] DAC Clock Generator Divisor. The DAC clock should be 500x the sample rate. For example, if bit [8] of this register is 0, and the PLL divide-by-two output is 72 MHz, a divisor of 9 would produce an 8 MHz clock to the DAC corresponding to a 16 KHz sample rate. If bit [8] is set, and the PLL divide-by-three output is 48 MHz, then a divisor of 3 would produce a 16 MHz clock to the DAC corresponding to a 32 KHz sample rate.

1-63 divide by 1 to 63
0 divide by 64

Bit [1] Set this bit *before* setting DAC enable in order to synchronize the *first* DAC frame with the next ADC end-of-frame pulse. The DAC runs thereafter at the rate determined by the divisor, bits [7:2].

Bit [0] DAC Enable
0 DAC disabled, reset, and powered-down
1 DAC enabled

dacFifo2 – DAC FIFO Register #2 (0xFC2D)

15	14	13	12	11	10	09	08
dacD15	dacD14	dacD13	dacD12	dacD11	dacD10	dacD09	dacD08
07	06	05	04	03	02	01	00
dacD07	dacD06	dacD05	dacD04	dacD03	dacC02	dacD01	dacD00

This register is used to write data to a holding register at the input to the DAC FIFO for channel #2. The channel #2 data will not be written into the FIFO until channel #1 is written.

dacFifo1 – DAC FIFO Register #1 (0xFC2E)

15	14	13	12	11	10	09	08
dacD15	dacD14	dacD13	dacD12	dacD11	dacD10	dacD09	dacD08
07	06	05	04	03	02	01	00
dacD07	dacD06	dacD05	dacD04	dacD03	dacC02	dacD01	dacD00

This register is used to write data to the DAC FIFO for channel #1. If the FIFO is full prior to a write operation, the data will be written and the oldest value in the FIFO will be discarded. Writing to this register will also transfer data from the DAC channel #2 holding register into the FIFO.

Reading this register will reset the FIFO block. A FIFO reset will set the FIFO to be empty. The data returned from the read will reflect the state of the DAC status register after the FIFO reset, with the IRQ and Underrun flags cleared.

dacStatus – DAC Status Register (0xFC2F)

15	14	13	12	11	10	09	08
dacfifo_nf	dacfifo_irq	dac_underrun					
07	06	05	04	03	02	01	00
					fcount2	fcount1	fcount0

bit [15] FIFO not full.

bit [14] IRQ is set whenever the number of empty locations in the FIFO exceeds the dacfid[2:0] field from bits [14:12] of the DAC Control Register. Cleared by writing any value this register.

bit [13] Error flag that is set if a DAC sample is missed because the FIFO was empty. Cleared by writing any value to this register.

bit [2:0] (R/O) Number of empty locations in the FIFO, 0 to 6. When the FIFO is empty, fcount[2:0] will be 6. This number is decremented when data is written to dacFifo1 and incremented when a sample is read from the FIFO every 500 DAC clocks.

This register will return 0x8006 when the DAC is reset (dacCtrl[0]=0).

Serial Communications

The NLP-5x provides these serial communication options:

1. A full-duplex UART.
2. A Synchronous Serial Port (SSP), supporting SPI master, SPI slave and I^2S formats.
3. A clock generator and modulator for sending and receiving infrared signals.
4. A USB 1.1 full-speed controller and physical interface.

Each of these blocks may be separately powered down.

UART

The NLP-5x includes a full-duplex UART. The transmit format is always 1 start bit, 8 data bits, and 2 stop bits. The receiver format is 1 start bit, 8 data bits, and a single stop bit.

UART I/O Pins

Pin	Name	Function
P2.7	TXD	UART serial data out. This pin is an output when bit 0 of uartCtrl is set high. It is a GPIO pin when uartCtrl[0] is low. This output is high when the UART is enabled, but idle.
P2.6	RXD	UART data in. This pin must be configured as an input, with or without a pullup device, when used for UART reception.

UART Clocking

The UART module is connected to the divide-by-2 output of the PLL, PLLCLK. A prescale function has four selections for dividing PLLCLK by 1, 2, 4 or 8. The output of the prescale function will be blocked when the UART is disabled, i.e., when uartCtrl[0] is zero. The output of the prescale function is the input to the fractional clock divider function. The purpose of the fractional divider is to generate an *average* clock rate that is an integral multiple of the standard baud rates. The fractional divider by its nature can introduce jitter (as much as one input clock period) in the UART clock. The output of the fractional divider is further divided down by the baud rate generator. The ultimate target UART clock is the baud rate x 16.

Example

In this example PLLCLK is 80 MHz, and the desired baud rate is 230400. The UART transmitter and receiver need a clock that is the baud rate x 16, so the desired UART clock (UCLK) is

Target UCLK = 230400 x 16 = 3.6864 MHz

Pick a multiplier, M, from 1 to 64 such that UCLK x M is close to, but less than, the PLLCLK rate. In this case M = 21, because

MCLK = 21 x 3.6864 MHz = 77.414 MHz.

M is the value that is loaded into bits [13:8] of the uartBaud register. If M is greater than 64, use the prescale bits uartBaud[15:14] to reduce the PLLCLK rate by factors of 2, 4 or 8.

Next, calculate an integer N, where N < = 256, such that

(PLLCLK/PS) * N / 256 = MCLK,

or

N = 256 * (MCLK / (PLLCLK/PS))

Where PS is the prescale factor (1, 2, 4 or 8).

In this example, N = 248 after rounding. N is the value that is written to the bits [7:0] of uartBaud. This value controls the fractional PLLCLK divider.

If N is programmed to be 248, and M is programmed to be 21, the average UART clock is thus:

UCLK = (248/256) * 80 MHz / 21 = 3.69 MHz.

The maximum jitter is one period of PLLCLK/PS.

uartBaud – UART Clock Selection Register (0xFC79)

15	14	13	12	11	10	09	08
preScale1	preScale0	baudDiv5	baudDiv4	baudDiv3	baudDiv2	baudDiv1	baudDiv0
07	06	05	04	03	02	01	00
frac7	frac6	frac5	frac4	frac3	frac2	frac1	frac0

cleared to zero on reset.

Bits [15:14] PLLCLK prescaler divider. These bits determine the clock rate that is input to the fractional divider.

0 0 PLLCLK
0 1 PLLCLK / 2
1 0 PLLCLK / 4
1 1 PLLCLK / 8

Bits[13:8] Baud Clock Divider. Divides the clock output of the fractional divider. The target rate is the desired Baud Rate x 16.

0 Fractional Divider Output / 64
1 Fractional Divider Output
…
63 Fractional Divider Output / 63

Bits [7:0] Fractional Divider
Determines average rate of clock input to Baud Clock Divider.
0 Fractional Divider Bypassed
1..255 Average rate is N/256 of output rate from prescaler divider.

uartCtrl – UART Control Register (0xFC78)

15	14	13	12	11	10	09	08
txReady	rxDone	rxOverrun	clrrxo	rxDoneMask			
07	06	05	04	03	02	01	00
			irqRefresh	txrMask	rxdMask	rxEnable	uartEnable

This register is cleared to zero by reset.

Bit [15] R/O 1: The transmit register is available for writing. This bit is cleared by writing to the uartData register (if uartEnable = 1) and set at the end of the second stop bit. This bit is also cleared if bit 0 of this register is zero.

This bit is OR'd into the UART interrupt request if bit 3 is set.

Bit [14] R/O 1: Set when a byte has been received. Cleared by reading the uartData register. This bit is also cleared if either bit 1 or bit 0 of this register is zero.

This bit is OR'd into the UART interrupt request if bit 2 is set.

Bit [13] R/O Sticky bit set if a new receive byte cannot be written into the receive hold register because it is already full. Cleared by writing a 1 to bit 12.

Bit [12] W/O Write a one to clear bit 13. Always reads zero.

Bit [11] R/O Zero while rxDone flag is blocked (held low) for a period after the uartData register has been read. Also zero if bit 0 or bit 1 of this register are zero.

Bit [4] W/O Writing a 1 to this bit will force the UART interrupt request low for 2 CPUCLK cycles. If the UART interrupt request is high, this will generate a positive going edge which is required by the processor. Normally only used in an interrupt service if both transmit and receiver interrupt sources are OR'd together.

Bit [3] 1: Bit 15 (txReady) is OR'd into the UART interrupt request.
0: Bit 15 (txReady) will not generate a UART interrupt request.

Bit [2] 1: Bit 14 (rxDone) is OR'd into the UART interrupt request.
0: Bit 15 (rxDone) will not generate a UART interrupt request.

Bit [1] 1: If bit 0 is also set, the UART receiver is enabled.
0: UART receiver disabled.

Bit [0] 1: UART clock generator enabled. Transmitter register enabled. Receiver will be enabled if bit 1 is also set.
0: UART clock generator disabled.

uartData – UART Transmit/Receive Data Register (0xFC7A)

Write

15	14	13	12	11	10	09	08
07	06	05	04	03	02	01	00
txData7	txData6	txData5	txData4	txData3	txData2	txData1	txData0

Writing to this register loads the data byte into the transmit hold register and starts transmission. Bit 15 of uartCtrl (txReady) is low while transmission is in progress. Software should not write to this register unless bit 15 of uartCtrl (txReady) is set.

Read

15	14	13	12	11	10	09	08
0	0	rxOverrun	0	0	0	parity	noStop
07	06	05	04	03	02	01	00
rxData7	rxData6	rxData5	rxData4	rxData3	rxData2	rxData1	rxData0

Software should not read this register unless bit 14 of uartCtrl is set (rxDone). This flag is set when a new byte is received and loaded into the receive hold register. When software reads uartData, the rxDone flag will be cleared.

Software must read the receive hold register before the next byte arrives, or the byte will be discarded and the overrun flag will be set.

Bit [13] Same as bit 13 of uartCtrl. Indicates an overrun happened.

Bit [9] Set if rxData[7:0] bits have an odd number of ones.

Bit [8] Set if the stop bit was missing, indicating a framing error.

Bits [7:0] Receive data byte.

Synchronous Serial Port (SSP)

The Synchronous Serial Interface supports a variety of serial protocols, including SPI Master, SPI Slave, and I2S Slave and Serial ROM Master. Up to four pins are used for the SSP functions. These four pins are general purpose I/O when not used by the SSP (that is, when sspCtrl[0] = 0).

The following table shows how pins P2[3:0] are used in the various SSP modes. If a pin is defined as an output in this table, it will become an output automatically, overriding the GPIO configuration bits in p2CtrlA and p2CtrlB for that pin. However, the converse is not true: in order for a pin in the table to function as an input, it must be configured to be an input (with or without a pull-up device) in registers p2CtrlA and p2CtrlB.

Synchronous Serial Port I/O Pins

Pin	Name	Function
P2.3	MOSI	SPI Master: Serial data out SPI Slave: Serial data in I2S: Serial data in Serial ROM: Serial data in/out
P2.2	MISO	SPI Master: Serial data in SPI Slave: Serial data out / Hi-Z I2S: Serial data out Serial ROM: GPIO
P2.1	SCLK	SPI Master: Serial clock out SPI Slave: Serial clock in I2S: Serial clock in Serial ROM: Serial clock out if sspCtrl[4] = 0
P2.0	SS_	SPI Master: GPIO if sspCtrl[4] = 0, otherwise active low slave select output SPI Slave: Active low slave select input I2S: Left/Right clock input Serial ROM: Serial clock out if sspCtrl[4] = 1

SPI Master Mode

In this mode the SSP is responsible for generating a clock signal that is output via pin P2.1. The clock rate is determined by sspCtrl[15:10]. The SPI master will produce one clock pulse for each bit transmitted. The word length can be 8, 16, 24 or 32 bits, determined by sspCtrl[9:8]. With SPI, transmission and reception occur simultaneously. The master device initiates the transmission/reception following a write to the sspData0 register (0xFC70).

If bit sspCtrl[4] is set, the SSP logic will drive the SS_ pin low prior to the first clock, and high after the last clock. This pin can be used to enable the slave device. If sspCtrl[4] is low, SS_ remains a GPIO pin and software can use this pin or other pin(s) to create the slave select signal(s).

The master device will shift data out of the MOSI pin while the slave device sends data to the MISO pin. Bit sspCtrl[7] determines whether the MSB or LSB is shifted out/in first. After the final clock, the SSP logic will set bit 15 of the sspStatus register. This bit can be used as an interrupt source. Software can clear this bit by writing any value to the sspStatus register.

For a word length of 8 bits, the shift data resides in sspData0[7:0]. The MSB is sspData0[7]. sspData1 will be zero when read, and sspData0[15:8] will be zero.

For a word length of 16 bits, the shift data resides in sspData0[15:0]. The MSB is sspData0[15]. sspData1 will be zero when read.

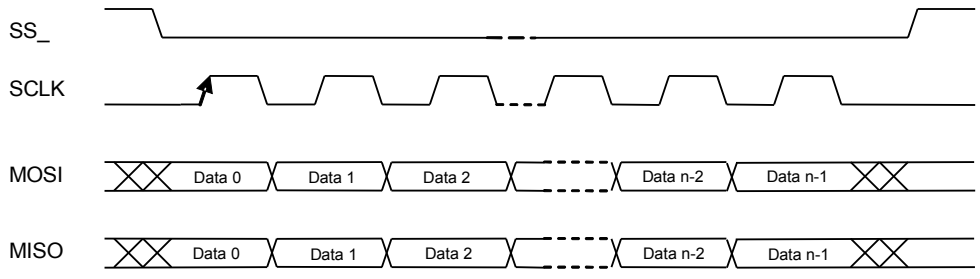
For a word length of 24 bits, the shift data resides in sspData1[7:0] and sspData0[15:0]. The MSB is sspData1[7]. sspData1[15:8] will be zero when read.

For a word length of 32 bits, the shift data resides in sspData1[15:0] and sspData0[15:0]. The MSB is sspData1[15].

SPI has four different operating modes selected by two bits in the SSP control register: CPOL and CPHA.
CPOL controls the polarity of the clock by specifying the inactive clock level. The clock is inactive between transmissions events, so CPOL specifies the clock level between transmission events. CPHA specifies the clock phase on which data is shifted in to the master and slave devices. If CPHA is zero, data is shifted in on the clock inactive-to-active edge. If CPHA is one, data is shifted in on the next clock edge, which is the active-to-inactive edge.

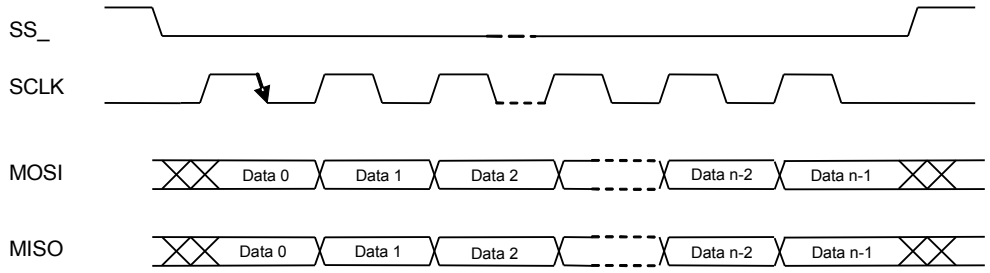
SPI Mode 0 (CPOL = 0, CPHA = 0)

Inactive clock polarity is LOW, data is clocked in on the first SCLK edge. In this mode both the master and slave must have the first data bit valid prior to the first clock. Note that the order of data shifting can either be LSB-first or MSB-first based on the setting of sspCtrl[7].



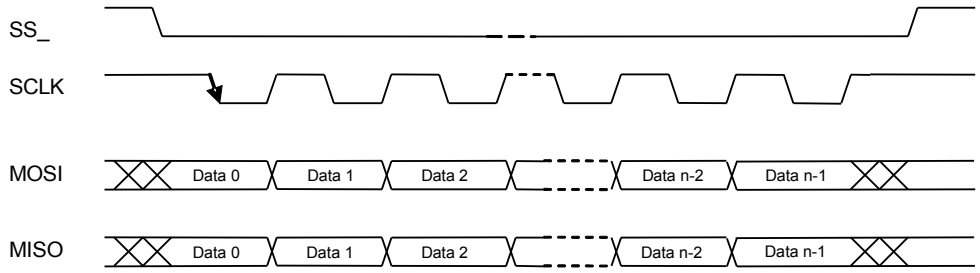
SPI Mode 1 (CPOL = 0, CPHA = 1)

Inactive clock polarity is LOW, data is clocked in on the second SCLK edge. In this mode the first data bit is available after the first SCLK clock edge. Note that the order of data shifting can either be LSB-first or MSB-first based on the setting of sspCtrl[7].



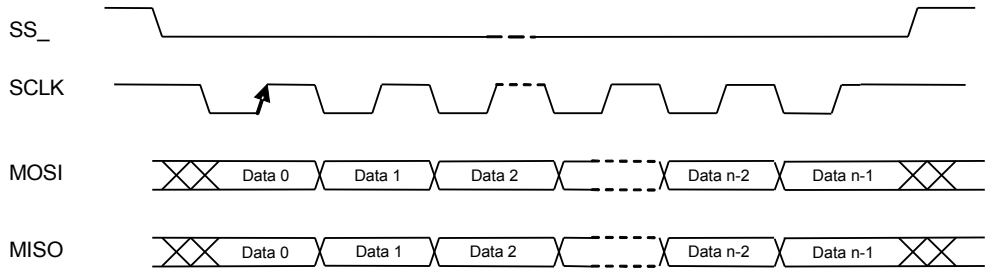
SPI Mode 2 (CPOL = 1, CPHA = 0)

Inactive clock polarity is HIGH, data is clocked in on the first SCLK edge. In this mode both the master and slave must have the first data bit valid prior to the first clock. Note that the order of data shifting can either be LSB-first or MSB-first based on the setting of sspCtrl[7].



SPI Mode 3 (CPOL = 1, CPHA = 1)

Inactive clock polarity is HIGH, data is clocked in on the second SCLK edge. In this mode the first data bit is available after the first SCLK clock edge. Note that the order of data shifting can either be LSB-first or MSB-first based on the setting of sspCtrl[7].



SPI Slave Mode

In this mode an external device is the master. The SCLK and SS_ pins become inputs, and the SSP clock generator is turned off. SS_ is used to select the NLP-5x as the current SPI slave. When SS_ is high, SCLK is ignored and the MISO pin is Hi-Z. When SS_ is low, MISO becomes an output pin.

The four SPI modes as described above are supported, as the option for LSB-first or MSB-first based on sspCtrl[7]. Word length can be specified to be 8, 16, 24 or 32 bits.

When the NLP-5x is required to transmit data to the master, the data must be written to the sspData0/sspData1 registers prior to the arrival of the first SCLK. The transmit register is double-buffered. Prior to the first SCLK, the shift register shadows the sspData0/sspData1 registers. After the first SCLK, the contents of the shift register are latched, so software can safely write new data to the sspData0/sspData1 registers. Bit 14 of the sspStatus register is a flag that indicates when the transmit shift register is latched. This flag is high when any data written to the sspData0/sspData1 registers will be directly copied to the transmit shift register, and it is low when the transmit shift register has been latched, and shifting is in progress.

After the final clock (as determined by the word length), the SSP logic will set bit 15 of the sspStatus register. This bit can be used as an interrupt source. Software can clear this bit by writing any value to the sspStatus register.

I2S Mode

I2S ("I Squared S") is a standard synchronous serial protocol for continuous streaming of stereo audio data from a host to a D/A converter, and/or from an A/D converter to a host. In I2S, the master device is responsible for generating the two required clock signals. The NLP-5x can only be an I2S slave, so the D/A, A/D, or CODEC device must be the master.

I2S has two clocks; the bit clock BCLK is typically 32 or 64 times the audio sample rate. It is used to shift data in and out. The LRCLK matches the sample rate and is used to indicate the start of a new left channel sample (when it goes from high-to-low) and the start of a new right channel sample (when it goes from low-to-high). The NLP-5x expects the bit clock to be connected to the SCLK pin, and the LRCLK to be connected to the SS_ pin. The MISO pin becomes an output for data to a D/A converter. The MOSI pin becomes an input for data received from an A/D converter.

Three word lengths are supported:

8-bit Left channel data will be shifted in and out of sspData0[7:0]. sspData0[15:8] will read zero.
Right channel data will be shifted in and out of sspData1[7:0]. sspData1[15:8] will read zero.

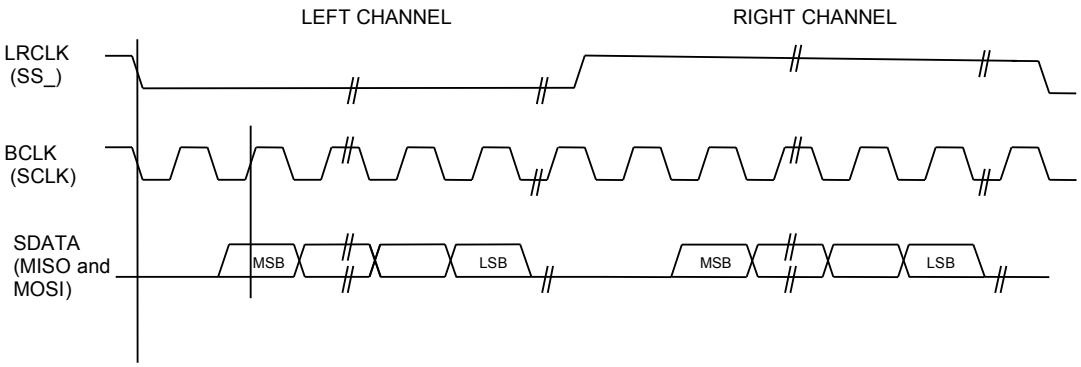
16-bit Left channel data will be shifted in and out of sspData0[15:0].
Right channel data will be shifted in and out of sspData1[15:0].

24-bit Left channel data will be shifted in and out of sspData0[15:0] and sspData2[15:8]. sspData2[15:8] are the 8 LSBs. sspData2[7:0] is ignored for writes and zero for reads.
Righ channel data will be shifted in and out of sspData1[15:0] and sspData3[15:8]. sspData3[15:8] are the 8 LSBs. sspData3[7:0] is ignored for writes and zero for reads.

The I2S master must provide at least the word-size number of BCLKs in each phase of LRCLK, but it can always have more BCLKs in each phase. The SSP logic stops shifting data in and out when the word-size number of bits is reached.

A transmit/receive frame begins when LRCLK goes from high-to-low. This signals the start of a new pair of left and right samples. The frame start is detected by the SSP logic, which transfers the current contents of the transmit hold registers to the transmit shift register. The SSP logic also sets bit 15 of the sspStatus register. This bit can be used as an interrupt source. Software can clear this bit by writing any value to the sspStatus register. The interrupt service routine can read the current contents of receive hold registers via peripheral ports sspData0-sspData3, and write new samples to the transmit hold register via the same ports.

The receive hold register is written from the receive shift register after the final bit of the right channel is received. Thus the software must read the receive hold register prior to the last bit of the right channel is received or a sample will be lost.


LEFT CHANNEL
RIGHT CHANNEL
LRCLK
(SS_)
BCLK
(SCLK)
SDATA
(MISO and
MOSI)
MSB
LSB
MSB
LSB

Serial ROM Mode

This mode works with a variety of devices that have a single, bi-directional data line between the host (master) and the slave device, and one or two clock lines.

Much of the information in the section on SPI Master mode applies to this mode, with some modification:

1) The MISO pin is not used for serial ROM mode. It remains a GPIO.

2) The MOSI pin can be an input or an output depending on the state of sspCtrl[2]. When MOSI is an output, the host is transmitting to the slave device. When MOSI is an input, the host is receiving from the slave device.

3) Some devices have two clock inputs, for example, one clock shifts in address data while the other shifts out memory data. In serial ROM mode the serial clock can be programmed to be an output on either the SCLK pin or the SS_ pin.

   If sspCtrl[4] is low, the SCLK pin is the serial clock output pin and the SS_ pin will be driven to the inactive clock polarity (sspCtrl[6]).

   If sspCtrl[4] is high, the SS_ pin is the serial clock output pin and the SCLK pin will be driven to the inactive clock polarity (sspCtrl[6]).

sspData0, sspData1, sspData2, sspData3 - Synchronous Serial Data Registers (0xFC70-0xFC73)

These four registers are used to write to the SSP Transmit Hold Register, which is write-only, and to read from the SSP Receive Hold Register, which is read only.

Both the transmit and receive register sets are double-buffered. This means that you can write new data to the transmit hold register during transmission of the current word, and that you can read the receive hold register while a new word is being shifted in.

sspData2 and sspData3 are only used for I2S 24-bit mode. In this mode sspData2[15:8] provide the 8 least-significant-bits of the 24-bit left channel data, and sspData3[15:8] provide the 8 least-significant-bits of the 24-bit right channel data. sspData3[7:0] is always read as zero.

In SPI master mode and serial ROM mode, a write to sspData0 is used to start the transmission and/or reception of a new word.

sspCtrl - Synchronous Serial Control Register (0xFC74)

15	14	13	12	11	10	09	08
sspps2	sspps1	sspps0	sspdiv2	sspdiv1	sspdiv0	wordLen1	wordLen0
07	06	05	04	03	02	01	00
msbf	cpol	cpha	ssmode	sspmode1	sspmode0	spimaster	sspon

Bits [15:13] Clock Prescale Divider
0 0 0 CPU Clock / 2
0 0 1 CPU Clock / 4
0 1 0 CPU Clock / 8
0 1 1 CPU Clock / 16
1 0 0 CPU Clock / 32
1 0 1 CPU Clock / 64
1 1 0 CPU Clock / 128
1 1 1 CPU Clock / 256

The SSP clock does not run when in SPI slave mode or I2S mode, or if sspOn = 0.

Bits [12:10] Clock Divider
0 0 0 SSP Prescale output / 8
0 0 1 SSP Prescale output / 1
.
.
1 1 1 SSP Prescale output / 7

The SSP clock does not run when in SPI slave mode or I2S mode, or if sspOn = 0.

Bits [9:8] Specifies the size of the word all modes.
0 0 8 bits
0 1 16 bits
1 0 24 bits
1 1 32 bits

Bit [7] 0 LSB is shifted out first and shifted in first
1 MSB is shifted out first and shifted in first

Note: this bit is ignored in I2S mode (always MSB first).

Bits [6:5] Select SPI Mode
CPOL selects the clock inactive polarity:
0 SCK is inactive low, active high
1 SCK is inactive high, active low
Where “inactive” means the state that SCK is in when no transmission is in progress. For example, if CPOL=0, then SCK will initially be low before transmission, and will return to a low state at the end of each transmission.

CPHA selects the clock “phase”:
0 data is shifted in on the first SCK edge after -SS goes low (start of new transmission). This edge will necessarily be a Positive going edge if CPOL=0 and a negative edge if CPOL=1. The first data bit must be ready prior to the first clock of a new transmission.
1 data is shifted in on the second SCK edge after –SS goes low (start o new transmission). This edge will necessarily be a negative going edge if CPOL=0 and a positive edge if CPOL=1. The first data bit will be shifted out on the first SCK edge after the start of a new transmission.

CPOL and CPHA together define the SPI Mode:

CPOL	CPHA	SPI Mode
0	0	0
0	1	1
1	0	2
1	1	3

These bits are ignored in I2S mode.

Bit [4] This bit controls the operation of the SS pin in SPI Master mode and serial ROM modes.

0: For SPI Master mode, the SS pin remains a GPIO.

For serial ROM modes, the SS pin is an output driven to the clock inactive polarity (bit 6), and SCLK is the serial clock output pin.

1: For SPI Master mode, the SS pin will be controlled automatically to be an output low during transmission, and return to an output high when transmission is complete.

For serial ROM modes, the SS pin will be the serial clock output pin and SCLK will be an output driven to the clock inactive polarity (bit 6).

Bits [3:2]
0 0 SPI mode, master or slave determined by bit 1.
0 1 I2S mode
1 0 serial ROM transmit mode
1 1 serial ROM receive mode

Bit [1] For SPI mode, selects master vs. slave:
0 SPI slave mode. SCK is an input.
1 SPI master mode. SCK is an output.

For I2S or serial ROM modes, this bit has no function

Bit [0] 0 Synchronous serial port disabled. P2.0-P2.3 are GPIO pins.
1 Synchronous serial port enabled.

sspStatus - Synchronous Serial Status Register (0xFC75)

15	14	13	12	11	10	09	08
sspirq	txlatch						
07	06	05	04	03	02	01	00

Bit [15] The SSP interrupt request flag is set in a manner depending on the SSP mode: For all modes other than I2S, the IRQ is set just after the receive data is copied into the receive hold register, signaling that reception and transmission of a word is complete.

For I2S mode the IRQ flag is set at the start of a new frame, just after the transmit hold register data is copied into the transmit shift register. This signals that the software can safely write the next pair of left and right samples to the transmit hold register.

The IRQ flag is cleared by writing any value to the sspStatus register, and it is cleared when the SSP is disabled (sspCtrl[0] = 0).

Bit [14] This signal may be useful in SPI slave modes. When high, the transmit shift register shadows the transmit hold register, so that any data written to the hold register is immediately copied to the shift register. This is the situation when the shift register is idle. When a new SCLK is received this signal will go low, isolating the shift register from the transmit hold register. The software can now write to the transmit hold register without the altering the contents of the transmit shift register.

Infrared Interface

The NLP-5x implements a half-duplex communications channel designed for use in simple, low-speed infrared applications. Infrared communication uses a serial data stream modulated at a carrier frequency. The NLP-5x will produce a modulated transmit signal, but it requires the receive signal to be base-band.

There are a large (and growing) number of IR data stream protocols. The NLP-5x will not directly support any specific data stream protocol; data stream interpretation will be left to the application. Instead, the NLP-5x will implement bit and pulse level control that is compatible with a variety of protocols. Each bit consists of a HIGH (IR present/modulated) period and a LOW (no IR present) period. The bit value (0 or 1) is determined by the sequence and/or durations of these periods, which vary according to the IR carrier frequency and the modulation protocol in use.

Receiver demodulation will be done by standard off-chip IR receiver HW modules. IO pin P2.4 is assigned as an input with no pullup device to accept the demodulated data when the IR Interface is enabled.

Transmission will use I/O pin P2.5, automatically assigned as an output when the IR Interface is enabled. The transmit LED will require an off-chip drive transistor.

When the IR Interface is disabled, IO pins P2.4 and P2.5 may be used as GPIO.

Terminology

The following equivalent terms are used in this specification:

Physical State	Signal Level	Logic condition
Modulated IR present	HIGH	on
IR not present	LOW	off

An IR bit normally consists of a sequence of two signal states, a *leading* state and a *trailing* state. During one of these states the signal level will be HIGH; during the other state the signal level will be LOW. The durations and order of the signal states determine the bit value (0/1) according to the specific protocol.

Features

The IR Interface provides these features:

- Support for carrier frequencies centered at about 40KHz, and ranging from 30KHz to 100KHz.
- Bit and pulse support compatible with bi-phase (such as RC5), pulse length and PWM data modulation formats.
- A " Complete" IRQ. When transmitting, this IRQ will be set when the bit has been completely transmitted. When receiving, this IRQ will flag a detected edge transition.
- Programmability of 0-bit-value and 1-bit-value pulse on/off (HIGH/LOW) durations with sufficient precision to allow transmission (but not reception) of edge- or phase-modulated data bits.
- Received pulse width (HIGH and LOW) reception measurement, allowing easy software determination of 0-bit, and 1-bit data values.
- Support for special bits (start/stop) similar to data bits.

The IR Interface consists of circuitry to generate high and low pulses of programmable duration, and to receive such pulses:

- Configuration control for selecting bit protocol details, carrier frequency, and transmit/receive mode.
- Registers that define the nominal 0 and 1 bit pulse on/off (HIGH/LOW) durations for data and start bits.
- Receive count registers, which contain the actual number of IRclk counts that are received for a HIGH (demodulated IR exists) and LOW (no demodulated IR exists).
- A IR status register.
- An IRQ, asserted on successful bit transmission, and on any received pulse edge transition.

Overview of Data Modulation Formats compatible with NLP-5x hardware support

The NLP-5x hardware supports bit transmission (a bit consists of HIGH and LOW pulse times). Transmission may be initiated with either a leading LOW or a leading HIGH period.

The NLP-5x supports received pulse time measurement. A pulse is considered the period between 2 alternating edges. For example, a HIGH pulse is defined as the period between a rising edge and a falling edge. The IRQ can be configured to respond to either, both, or no transition edges. A pulse period is measured as the number of irClk cycles that occur in that period (irClk frequency is set in IRconfig[11:0]).

It is the software's responsibility to assemble received pulse, or transmitted bit data into a data stream. It is also software's responsibility to determine any received error conditions, such as short HIGH pulse.

Bi-phase format is an edge based, Manchester-type encoding with a fixed bit transmission period (1.778mS). The most common example is "RC5". A 1 is defined as a low to high transition, nominally half way through the bit period. A 0 is defined as a high to low transition.

PWM format has a fixed bit cycle time, but a variable duty cycle. For example a 25% High / 75% LOW may indicate a "1", while a 75% HIGH / 25% LOW would indicate a "0".

Pulse Length format has a variable bit cycle time. For example, a "1" may consist of 1.2mS HIGH and .6mS LOW. A "0" might consist of .6mS for both LOW and HIGH.

irConfig – IR Configuration (0xFC50)

15	14	13	12	11	10	09	08
irEnable	xmit	leadingHigh	invertRcv	irClk11	irClk10	irClk09	irClk08
07	06	05	04	03	02	01	00
irClk07	irClk06	irClk05	irClk04	irClk03	irClk02	irClk01	irClk00

cleared to zero on reset.

Bit [15] IR Enable (R/W)
0: IR interface is powered down.
1: IR interface is powered-up and operates.
NOTE: setting this bit assigns and configures P2.4 as the IR input and P2.5 as the IR output.

Bit [14] IR Transmit (R/W)
0 IR interface is in receive state
1 IR interface is in transmit state
NOTE: since the IR I/F is half-duplex, portions of the non-selected subsystem may be powered down if easily feasible.

Bit [13] Leading High pulse (R/W)
0 Leading pulse is LOW; trailing pulse is HIGH
1 Leading pulse is HIGH; trailing pulse is LOW

Bit [12] Invert Receive (R/W)
0 IR presence indicated by a HIGH on P2.4
1 IR presence indicated by a LOW on P2.4.

This bit, when set, inverts the polarity of the P2.4 input to the IR logic to accommodate IR demodulators with active low outputs.

Bits [11:0] IR Carrier Clock Frequency (R/W). The IR carrier clock is derived from CLK1 (nominal 40 MHz). The divisor specifies a countdown value that determines the carrier frequency (range from 9765 Hz through 40 MHz). The carrier clock is the timing source for the rest of the IR interface.

1-4095 divide by 1 to 4095
0 divide by 4096

irStatus – IR Status (0xFC51)

15	14	13	12	11	10	09	08
complete	start_trans	p_edge	n_edge	rcvdEdge	rcvState		
07	06	05	04	03	02	01	00

cleared to zero on reset.

Bit [15] IR Complete (R/W).
0: IR bit transmission/reception in process or not started
1: IR bit transmission/reception complete
NOTE: This bit also sets the IR IRQ condition. On receive this bit is set based on the state of irStatus[13:12].
Software can clear this bit by writing a zero.

Bit [14] IR Start Transmission (W/O)
0: any write to this bit starts a bit transmission if irCfg[14] is set
1: any write to this bit starts a bit transmission if irCfg[14] is set
NOTE: the value of this bit is don't-care; The 0/1 state of the transmitted bit is determined by the pulse duration registers and the leadingHigh bit.

Bit [13] IR p_edge (R/W)
0: IR receive positive edge transition will not trigger IRQ
1: IR receive positive edge transition will trigger IRQ

Bit [12] IR n_edge (R/W)
0: IR receive negative edge transition will not trigger IRQ
1: IR receive negative edge transition will trigger IRQ

Bit [11] IR received edge (R/O)
0: IR IRQ was triggered by negative edge transition (HIGH/LOW)
1: IR IRQ was triggered by positive edge transition (LOW/HIGH)
NOTE: valid only in receive mode when bit[15] is 1.

Bit [10] IR Receive Status (R/O)
0: Demodulator input signal is LOW
1: Demodulato input signal is HIGH
NOTE: This bit reflects the input state (P2.4) at all times when the IR Interface is enabled. This allows a completely polled receive if bits [13:12] are clear.

Definition of Registers 0xFC52-3 (IR Transmit Pulse Duration Registers)

These two registers define the countdown values for the transmitted LOW and HIGH states. These registers must be programmed as required for the "0" and "1" bit values. Each register contains a 16-bit countdown value in units of the IR Carrier Clock. During transmission these register values determine the transmitted pulse durations. The transmission order is determined by irCfg[13]. A 0 value in either register means that pulse is of 0 duration. For example, if the trailing register was 0, the bit-complete IRQ will be asserted at conclusion of the leading register. Alternately, if the leading register is 0, the trailing register will immediately become the active phase. These registers are unused during reception.

The register names and addresses are:

Register	Address	Controls
IRXmtCntLow	0xFC52	duration of LOW pulse
IXmtCntHigh	0xFC53	duration of HIGH pulse

The prototype Pulse Duration Register is:

15	14	13	12	11	10	09	08
count15							
07	06	05	04	03	02	01	00
							count00

cleared to zero on reset.

Definition of Registers 0xFC54-55 (IR Receive Count Registers) (R/C)

These two registers contain the actual received duration values for the LOW and HIGH signal states. Each register contains a 16-bit up-count value clocked by the IR Carrier Clock. On a rising edge transition, RcvCntHigh is cleared, and begins counting. RcvCntLow value is held. On a falling edge, RcvCntLow is cleared and begins counting. RcvCntHigh value is held. During transmission these registers are not used.
These registers are cleared by writing any value.

The register names and addresses are:

Register	Address	Contents
IR RcvCntLow	0xFC54	actual duration of received LOW pulse
IR RcvCntHigh	0xFC55	actual duration of received HIGH pulse

The prototype RcvCntXxx Register is:

15	14	13	12	11	10	09	08
count15							
07	06	05	04	03	02	01	00
							count00

cleared to zero on reset.

Protocol for Transmission and Reception

Sending/receiving a protocol word with the IR Interface consists of a number of separate programming steps:

a) Setting up the protocol start condition
b) Sending or receiving the start bit
c) Setting up the protocol data bit conditions
d) Sending each bit of the data word as a separate transaction – could be 14 or more bits depending on protocol. Alternately, receiving and parsing the pulse data into bits of the bit stream.
e) Depending on the protocol, sending or receiving a stop bit

Bit transmission is initiated by writing any value to Bit 14 of the IR status register. It will be the user's responsibility to configure the pulse duration and irCfg registers before initiating a transmission. .Note: the other Writable bits of IR Status are used only for receive, so their values are also "don't-care" for a write.

Receiving an IR Bit under any protocol

The (0/1) logical state of a received bit is not determined by the IR Interface but must be determined by the software program for all protocols. Here is an illustration of this process: We have defined HIGH to mean an active IR pulse. To receive one IR bit, first the irClk rate is set, the irCfg enables IR and is set to receive mode with the p_edge IRQ bit set. The RcvCntLow begins to count. We ignore the value in this register, and wait until the IRQ, indicating the onset of active IR modulation. At that point, we set the n_edge IRQ bit, and at the next IRQ read the RcvCntHigh register. If software determines this is a legal start bit's HIGH pulse (i.e., its duration is within specification for the protocol), continue to monitor the RcvCntx registers for the appropriate follow on pulses (more start or data bits). With the IR interrupt enabled for both edges, the IR IRQ routine may test the rcvdEdge bit of the IRStatus register to determine the detected edge.

Typically the protocol uses different low/high durations for special bits, compared to data bits. These special durations and tolerances are also parsed in software.

Transmitting an IR Bit in PWM and Pulse-length modes

The Pulse Duration registers are programmed with the nominal carrier clock counts for each state, irCfg[13] (the leading edge bit) is configured, and the IR Status register is written with any value (keeping in mind that this can affect the state of the x_edge receive-mode IRQ bits). This starts the transmission of the values in the 2 XmtXxx registers, with order based on the IRCfg[LeadingHigh] bit. Writing to the IR Status register:

1) Places the transmit subsystem into LOW state (or HIGH, if leadingHigh bit, 0xFC50[13], is set).
2) Clears an internal register that counts up, clocked by the carrier clock.
3) Digitally modulates the IR output by the carrier clock if in the HIGH state.

When the transmission clock reaches the count contained in the appropriate Pulse Duration Register (as determined by the state of the leadingHigh bit), the output is switched to HIGH (LOW if leadingHigh is true) and the internal counter is cleared and starts counting up again, clocked by the carrier clock. When the internal counter reaches the appropriate duration count, the IR IRQ is asserted with the complete bit set (0xFC51[15]=1). It is the software's responsibility to turn off the IR drive if the bit ends with the drive HIGH and the protocol requires it to return to LOW.

The values in both Pulse duration registers are transferred to internal counters for the bit transmission. The values in the duration registers may therefore be modified (or retained) as soon as a transmission starts in preparation for the next bit.

Alternate operation

Since the receive mode is edge triggered, it may be used as an additional I/O irq (P2.4) if IR is not used. Similarly, the transmit mode may be used as an additional timer based IRQ if care is given to the state of P2.5

USB 1.1 Interface

The NLP-5x has a full-speed USB device controller. Details of this block are in a separate document. Documented here is the mechanism for communicating with the USB controller.

The USB block requires a 48 MHz clock. It is recommended that the PLL be programmed to generate a VCO frequency of 144 MHz, and that the VCO/3 output (48 MHz) be selected for the USB clock. The VCO/3 tap is selected for the USB clock by setting bit 4 of the clock select register, clkSelect. The CPU clock should be selected to be 72 MHz (turbo mode) or 36 MHz (non-turbo mode).

Since the processor and the USB block are in different clock domains, communication is facilitated by a set of buffers and state-machine logic to transfer a block of 1 to 8 bytes from the buffers to the USB device (write transfer), or from the USB device to the buffers (read transfer). The USB block uses an 8-bit-wide data bus, so data is transferred a byte at a time.

There are separate buffer pairs for writing to the USB device and reading from the USB device. In the write direction, there is a pair of 8-byte buffers arranged as 4 x 16 bits. The individual buffers are called the “Write A Buffer” and the “Write B Buffer”. The processor can write directly into the Write A Buffer via peripheral registers usbData0-usbData3. When a write transfer is started via a command written to the usbCmd register, the contents of the A buffer are copied to the B buffer, and the transfer operation will read bytes from the B buffer when sending them to the USB device. The bytes are transferred in order usbData0[7:0], usbData0[15:8], usbData1[7:0] etc. This “double buffering” allows the processor to write to the A buffer without disturbing a transfer in progress.

In the read direction, there is a second pair of 8-byte buffers arranged as 4 x 16 bits. These two buffers are called the “Read A Buffer” and the “Read B Buffer”. When data is transferred from the USB device, a byte at a time, it is written into the Read A Buffer. With the exception of the first byte of Read Buffer A, the processor cannot directly read the A buffer. After the read transfer is complete, the contents of the A buffer will be copied to the B buffer by another command from the processor written to usbCmd. Then the processor can read the data from the B buffer via usbData0-usbData3 registers.

A busy flag can be polled to determine when a transfer operation has completed.

The USB controller can have many interrupt sources with separate enable flags. If the interrupt handler for the USB interrupt does not service and clear all the active interrupt flags, the interrupt service routine would exit with the USB interrupt request still high. Since no edge is generated, a new interrupt will not happen. A command written to register usbCmd can be used in this case to force the USB interrupt request line low for a few processor clocks. This will generate a positive going edge on the USB interrupt request line if it is currently high, and register a new interrupt request.

Configuring the USB Physical Interface

The USB controller can support two different types of USB physical interface signalling modes. To correctly configure the USB controller to match the USB physical interface, software must clear bit 4 of the USBSTS register (register 16H) within the USB controller prior to using the USB physical interface as an output. The reset state for bit 4 is high, which is incorrect.

usbCtrl – USB Control/Status Register (0xFC5A)

15	14	13	12	11	10	09	08
irq		suspend	xmit				
07	06	05	04	03	02	01	00
reserved (0)	reserved(0)	usbRam0	xcvr_sp	xcvr_on	reserved	resume	-reset

cleared to zero on reset.

Bit [15] (R/O) Interrupt signal from USB block.

Bit [14] Reserved.

Bit [13] (R/O) Read-only suspend state signal from USB block. This signal, when low, will cause a wake event if bit 12 of wakeMask is set.

Bit [12] (R/O) USB transmit active flag

Bit [7:6] Reserved. *Must be written zero*.

Bit [5] 0 256x16 RAM is not connected to the USB block but is accessible for reading and writing by programs at addresses 0xFD00-0xFDFF

1 256x16 RAM is connected to the USB block for endpoint #0 and #1. Programs running on the NLP-5x cannot directly read or write this RAM.

Bit [4] Transceiver speed control.
0 low speed
1 full speed (required)

Bit [3] Transceiver enable (required). If zero, the transceiver is Hi-Z.

Bit [2] Reserved. Must write as zero.

Bit [1] Resume signal: set high to wake the USB block from suspend state.

Bit [0] 0: Hold USB block in reset.
1: Release USB block from reset.

usbCmd – USB Command Register (0xFC5B)

This register behaves differently for writes than it does for reads. The write and read functions are described separately.

Write

15	14	13	12	11	10	09	08
					cnt2	cnt1	cnt0
07	06	05	04	03	02	01	00
cmd1	cmd0	postinc	adr4	adr3	adr2	adr1	adr0

This register is used to initiate transfer of data from the buffers addressed via usbData3-usbData0 registers to the USB block, or from the USB block to buffers addressed via usbData3-usbData0 registers.

Bits [10:8] Transfer size in bytes, minus one.
0 0 0 Transfer 1 byte
0 0 1 Transfer 2 bytes
…
1 1 1 Transfer 8 bytes

Bits [7:6] Command

0 0 Force the USB interrupt request line low for 2 processor clocks.
Used to generate a positive edge if the interrupt request is currently high.
0 1 Transfer write buffer A to write buffer B, followed by transfer
of data from write buffer B to the USB block.
1 0 Transfer read buffer A to read buffer B, followed by transfer
of data from the USB block to read buffer A.
1 1 Transfer read buffer A to read buffer B, no transfer to/from USB block.

A command (other than 00) will not start if usbCtrl[0] is clear, and a command in progress will be aborted if usbCtrl[0] is set low.

If the command is 00, all the other bits of this register are don't-care.

Bit [5] 0: register address is unchanged during transfer.
1: increment register address after each read/write

Bits [4:0] USB register to read or write.

Read

15	14	13	12	11	10	09	08
busy	0	0	0	0	0	0	0
07	06	05	04	03	02	01	00
rdata7	rdata6	rdata5	rdata4	rdata3	rdata2	rdata1	rdata0

Bit [15] Read-only flag that a transfer is in progress. Set high by any write to this register with either bit 7 or bit 6 set. Cleared when the transfer is complete or if usbCtrl[0] is clear.

Bits [14:8] All zeros

Bits [7:0] Current contents of least signification byte of read buffer A.

LCD Interface

The NLP-5x can drive an external static or multiplexed Liquid Crystal Display (LCD) with up to 104 segments using specially configured I/O pins. The LCD interface has these features:

- Configurable outputs, up to 26x4
- Programmable drive control handles display refreshing without software intervention and allows operation (maintaining a display with unchanging information) while the processor is stopped
- Programmable refresh rate
- Static, 1/2, and 1/3 LCD bias
- Low 80 µA current consumption

The LCD interface consists of circuitry to generate pulse sequences of varying amplitude supplied to a programmable number of outputs over a period of programmable duration. Each pixel in the LCD is supplied by two pulse sequences, a “common” and a “segment”. Pixels in the LCD that are to be “on” (dark) are driven by a pair of pulse sequences that produce a large AC voltage; pixels that are to be “off” (transparent) are driven by a pair of pulse sequences that produce a small-or-zero AC voltage. To protect the LCD itself, no pixel may have a net DC voltage. The application software needs only to control the on/off state of each pixel; the pulse sequences are generated automatically by the LCD interface.

The LCD interface circuitry includes:

- Configuration control and I/O assignment registers
- 128-bit LCD pixel memory
- Programmable refresh clock
- state machine to control timing waveforms
- external connections through I/O to segments, commons, and bias voltages

Using the LCD interface requires dedicating some number of I/O pins for that purpose. The largest configuration requires all 32 bits of P0 and P1. Two LCD I/O assignment registers define which bits are for LCD use. Due to the necessity of maintaining zero net DC voltage across the LCD pixels, any I/O pins used for LCD may be for any other purpose only with caution.

LCD Registers

The LCD interface contains a control register, two I/O assignment registers, and eight data registers.

lcdCtrl – LCD Control Register (0xFC40)

15	14	13	12	11	10	09	08
lcdEnable					2_3Cbias	commons1	commons0
07	06	05	04	03	02	01	00
lcdSync	lcdClk6	lcdClk5	lcdClk4	lcdClk3	lcdClk2	lcdClk1	lcdClk0

cleared to zero on reset.

Bit [15] LCD Enable (R/W)
1: LCD output waveforms are produced.
0: LCD interface is powered down, no waveforms are produced.
NOTE: OSC2 or OSC3 must be separately enabled for LCD operation.

Bit [10] 2_3 Commons bias mode (R/W)
0 1/2 Bias
1 1/3 Bias
(see also bits[9:8])

Bits [9:8] Commons select. (R/W) – determines the number of commons outputs.
0 0 1 (always static bias)
0 1 2 (bias from bit10)
1 0 3 (bias from bit10)
1 1 4 (always 1/3 bias)

Bit [7] LCD Sync (R/C)
1: Start of Frame detected
0: Start of frame not detected
This bit may be read or cleared by software. Is set only by hardware. This bit can be used as an interrupt source.

Bits [6:0] LCD Clock Generator Divisor. (R/W) The LCD clock is derived from one of the low speed oscillators divided by 8 (nominal 4096 Hz). The divisor specifies a further reduction in the LCD clock.

1-127 divide by 1 to 127
0 divide by 128

The LCD Sync bit is set at the beginning edge of the common COM0 time frame. The bit may be polled and cleared in software.

LCD I/O Assignment Registers

Bits are set in the LCD I/O Assignment registers to designate I/O pins that are to be used with the LCD interface. Bits left clear are not used for LCD and may be used for other I/O purposes. As shown below the relationship between specific I/O pins and LCD segments/commons is fixed for each of the four common configurations. The “lcdxx” signal designation is used to relate the segments and commons to the actual I/O pins.

For correct operation, each P0 or P1 pin that is configured for LCD use must be programmed to be an input with no pull-up device using the p0CtrlA/B and p1CtrlA/B registers, respectively. Even though a port pin is configured as an input, the input buffer will be disabled when a pin is configured for LCD use.

To configure a port pin as an input with no pull-up, the p0CtrlB (or p1CtrlB) bit associated with the port pin should be set to one, while the p0CtrlA (or p1CtrlA) bit should be zero.

lcdP0Assign – LCD P0 Assignment Register (0xFC42)

This register specifies which bits of Port P0 are assigned to LCD functions.

15	14	13	12	11	10	09	08
lcd15	lcd14	lcd13	lcd12	lcd11	lcd10	lcd09	lcd08
07	06	05	04	03	02	01	00
lcd07	lcd06	lcd05	lcd04	lcd03	lcd02	lcd01	lcd00

cleared to zero on reset.

lcdP1Assign – LCD P1 Assignment Register (0xFC43)

This register specifies which bits of Port P1 are assigned to LCD functions.

15	14	13	12	11	10	09	08
lcd31*	lcd30*	lcd29*	lcd28*	lcd27*	lcd26*	lcd25	lcd24
07	06	05	04	03	02	01	00
lcd23	lcd22	lcd21	lcd20	lcd19	lcd18	lcd17	lcd16

* may be overridden by contents of LCD Control Register

cleared to zero on reset.

Mapping of segments and commons to lcd IO bits

The tables below show the assignment of each of the 32 lcd signals to a specific IO pin as a function of the configuration. Segments are designated as “sxx” (xx= 00-30) and commons are designated as “cx” (x= 0-3). The commons pins are automatically assigned based on the contents of LCD Control Register [9:8] and may not be overridden by software. The “bias1” and “bias0” pins are assigned based on the contents of LCD Control Register [10:8] and may not be overridden by software.

STATIC MODE (one common, COM0)

lcd31	lcd30	lcd29	lcd28	lcd27	lcd26	lcd25	lcd24	lcd23	lcd22	lcd21	lcd20	lcd19	lcd18	lcd17	lcd16
c0	s30	s29	s28	s27	s26	s25	s24	s23	s22	s21	s20	s19	s18	s17	s16
lcd15	lcd14	lcd13	lcd12	lcd11	lcd10	lcd09	lcd08	lcd07	lcd06	lcd05	lcd04	lcd03	lcd02	lcd01	lcd00
s15	s14	s13	s12	s11	s10	s09	s08	s07	s06	s05	s04	s03	s02	s01	s00

Maximum configuration is 31 pixels (31x1).

MUX2 MODE (two commons, COM1, COM0)

lcd31	lcd30	lcd29	lcd28	lcd27	lcd26	lcd25	lcd24	lcd23	lcd22	lcd21	lcd20	lcd19	lcd18	lcd17	lcd16
bias1	bias0	c1	c0	s27	s26	s25	s24	s23	s22	s21	s20	s19	s18	s17	s16
lcd15	lcd14	lcd13	lcd12	lcd11	lcd10	lcd09	lcd08	lcd07	lcd06	lcd05	lcd04	lcd03	lcd02	lcd01	lcd00
s15	s14	s13	s12	s11	s10	s09	s08	s07	s06	s05	s04	s03	s02	s01	s00

Maximum configuration is 56 pixels (28x2).

MUX3 MODE (three commons, COM2, COM1, COM0)

lcd31	lcd30	lcd29	lcd28	lcd27	lcd26	lcd25	lcd24	lcd23	lcd22	lcd21	lcd20	lcd19	lcd18	lcd17	lcd16
bias1	bias0	c2	c1	c0	s26	s25	s24	s23	s22	s21	s20	s19	s18	s17	s16
lcd15	lcd14	lcd13	lcd12	lcd11	lcd10	lcd09	lcd08	lcd07	lcd06	lcd05	lcd04	lcd03	lcd02	lcd01	lcd00
s15	s14	s13	s12	s11	s10	s09	s08	s07	s06	s05	s04	s03	s02	s01	s00

Maximum configuration is 81 pixels (27x3).

MUX4 MODE (four commons, COM3, COM2, COM1, COM0)

lcd31	lcd30	lcd29	lcd28	lcd27	lcd26	lcd25	lcd24	lcd23	lcd22	lcd21	lcd20	lcd19	lcd18	lcd17	lcd16
bias1	bias0	c3	c2	c1	c0	s25	s24	s23	s22	s21	s20	s19	s18	s17	s16
lcd15	lcd14	lcd13	lcd12	lcd11	lcd10	lcd09	lcd08	lcd07	lcd06	lcd05	lcd04	lcd03	lcd02	lcd01	lcd00
s15	s14	s13	s12	s11	s10	s09	s08	s07	s06	s05	s04	s03	s02	s01	s00

Maximum configuration is 104 pixels (26x4).

Examples:

In MUX4 mode, COM3-COM0 are assigned to P1.13-P1.10, and SEG7 is assigned to P0.7.
In MUX2 mode, COM1-COM0 and assigned to P1.13-P1.12, and SEG7 is assigned to P0.7.
In static mode, COM0 is assigned to P1.15 and SEG7 is assigned to P0.7.

LCD Data Registers

Eight data registers comprising 128 bits may be programmed with the pixel content for the LCD. Each bit of a data register corresponds to one pixel in the LCD. A *cleared* bit produces a transparent pixel; a *set* bit produces a dark pixel.

The table shows the relationship between segments, commons, and the pixel addresses in the data registers. A pair of registers provides the pixel content for one common.

	COMMONS			
segments	COM0	COM1	COM2	COM3
0-15	lcdData0	lcdData2	lcdData4	lcdData6
16-31	lcdData1	lcdData3	lcdData5	lcdData7

Definition of Registers 0xFC44-0xFC4B (lcdData0-lcdData7)) (W/O, R/O)

All cleared to zero on reset.

Within each data register, segments and commons are related to register bits as:

LCDDATA[register n][bit b] = COM(2n), SEG(b+16*(n:0))

(n=0-7, b=0-15)

For example, LCDDATA0[4] contains the pixel for COM0, SEG4;
LCDDATA3[9] contains the pixel for COM1, SEG25

It is suggested that the LCD Data Registers be implemented as D-type FFs, write-only by the processor, and read-only by the LCD waveform control circuitry. Memory writes will not interfere with LCD fetch and refresh timing because the F/Fs read and write ports are separate. Since there is no hardware synchronization between the Data Register write and LCD drive, there is the possibility that any one segment may be out of date by 1 frame. This small time period error should not cause any visible artifacts. It is possible that a Data Register is written to at the same time that its output is refreshing the display. Therefore, the commons selector may need a latch to ensure stable data is refreshing the column. Software use of the LCDSync bit may help minimize such artifacts.

The Data Registers must be readable when the processor is in IDLE mode.

Bits in LCD Data Registers that are not assigned to an IO pin via the LCD I/O Assignment Registers are “don’t care”.

LCD Clock Control

The clock source for the LCD Interface is one of the two low speed oscillators (crystal or RC), nominally 32768 Hz. Bit 3 of the low speed oscillator control register (#IsoCtrl) is used to select the oscillator source for the LCD function. The selected oscillator must be enabled. The selected clock is first divided by 8, and then further divided by the LCD Clock Generator Divisor to give and LCD Clock rate between 4096 Hz and 32 Hz. The rate can be selected to minimize flicker under fluorescents at different line frequencies. The lowest rate could be used as the slowest static mode clock.
.

During IDLE mode the LCD Interface may continue to provide waveforms to the LCD. Because the processor is stopped in this state, the pixel information may not change during IDLE.

When the LCD Interface is disabled (by clearing bit15 of register 0xFC40), clocks are disabled, circuitry is powered down, and any I/O Pins programmed for LCD operation revert to the conditions programmed in the PIO port configuration registers. That is, the LCD I/O Assigns register is gated by the LCD enable bit. It seems unlikely the PIO configuration for pins also used by the LCD would ever be anything but the weak pullup, unless the app developer had a real good reason to modify the I/O configs. If the developer had a good reason, and was aware of the DC issues (or other conflicts), it would be his responsibility to properly handle the LCD.

LCD Bias and Waveform Generation

In the static mode of operation, the COM0 and SEGn signals pulse between LCDVDD and LCDVSS. In other modes the LCD Interface must produce complex output waveforms that include voltages at levels other than VDD and VSS. In “1/2-bias”, the waveforms have 3 levels; in “1/3-bias” the waveforms have 4 levels. Two of these levels are LCDVDD and LCDVSS; the other levels are obtained from I/O pins P1.15 and P1.14, which are typically connected to an external resistor ladder network.

In 1/2-bias, P1.15 and P1.14 are connected together and to an external voltage that is nominally LCDVDD/2. In 1/3-bias, P1.15 is connected to an external voltage that is nominally 2*LCDVDD/3 and P1.14 is connected to an external voltage that is nominally LCDVDD/3.

The complex waveforms produced by the LCD Interface and presented to the COM and SEG pins must have these characteristics:

- transparent pixels must have a minimum or zero AC net voltage
- dark pixels must have a maximum AC net voltage
- no pixel may have a net DC voltage when averaged over a time period of one frame.

Waveform control values may be developed for all memory bits; only the bits enabled by the LCD I/O Assignment Registers will actually be output.

Motor Controllers

The NLP-5x motor control block provides 3 bi-directional pulse-width-modulator (PWM) units, and 3 interrupter-based tachometer sensor units.

The sensor section supports either quadrature-encoded or single-ended tachometer input signals for each sensor unit. Each unit contains a 16-bit forward-direction and a 16-bit reverse-direction tachometer pulse counter. In single-ended mode, only the forward-direction counter is used. Each counter also has an associated overflow bit. This provides a 17-bit counter range.

PWM Section

The PWM section provides up to 3 independent pulse-width-modulators, each with 9 bits of resolution (8 magnitude bits and 1 polarity bit). These units have independent periods and duty cycles

Each PWM unit that is enabled is assigned two I/O pins of port P1. The I/O pins must be configured as outputs (rather than inputs) by the application. These outputs have standard GPIO drive capability and are intended to drive external H-bridges, which will provide the power to the motors.

Each PWM unit has a phase bit (mpwmReload*n*[14]) that determines if the PWM outputs are active-high or active-low. For a given PWM unit, if the phase bit is *clear*, one output will generate high-going pulses, while the other is held low. In other words, the output pulse is *active-high*. If a unit's phase bit is *set*, one output will generate low-going pulses, while the other is held high. In other words, the output pulse is *active-low*.

The phase bit is provided because the NLP-5x resets with port pins configured as inputs with weak pull-up devices. If the external H-bridge activates a high-side/low-side pair based on an high level on a PWM output pin, then a high level on *both* PWM pins has the potential to activate *both* high-side/low-side pairs. This results in a short-circuit. Normally, this situation is handled in one of two ways:

a. This state (both outputs high) is recognized and handled in the external H-bridge circuitry to avoid the short. Often this state is used to create a "braking" function.

b. Both PWM output pins are pulled low with external resistors to prevent the H-bridge from turning on at the NLP-5x reset.

Most H-bridge designs work with an active high input turning on the high-side/low-side pair, and properly handle this condition as above.

Each PWM unit has a polarity bit (mpwmReload*n*[15]) that determines which of the two outputs is held steady while the other generates pulses. This is how an H-Bridge would be driven for forward or reverse motor motion. This is a magnitude and sign format, not a 2's-complement signed format. If the polarity bit is *clear*, the motor is defined as going in a *forward* direction, and if the polarity bit is *set*, the motor is defined as going in a *reverse* direction.

.

PWM I/O Assignment

Motor PWM #0 Polarity = 0 (Forward)	P1.8
Motor PWM #0 Polarity = 1 (Reverse)	P1.9
Motor PWM #1 Polarity = 0 (Forward)	P1.6
Motor PWM #1 Polarity = 1 (Reverse)	P1.7
Motor PWM #2 Polarity = 0 (Forward)	P1.4
Motor PWM #2 Polarity = 1 (Reverse)	P1.5
All-Stop Input	P1.1

PWM Registers

There are 2 registers for each PWM unit plus one register that determines linkage between units. Here is the assignment of register addresses for the PWM registers:

mpwmCtrl0	0xFCC0
mpwmReload0	0xFCC1
mpwmCtrl1	0xFCC4
mpwmReload1	0xFCC5
mpwmCtrl2	0xFCC8
mpwmReload2	0xFCC9
mpwmLink	0xFCD0

mpwmCtlr0 - mpwmCtrl2 (0xFCC0, 0xFCC4, 0xFCC8)

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
enable	mpwm_clkdiv														

cleared to zero on reset.

Bit [15] 1: PWM unit outputs will drive assigned GPIO pins if those pins are configured as outputs.
0: PWM unit outputs will not be connected to the assigned GPIO pins.

Bits [14:0] The PWM clock rate generator is a 15-bit divider from the system clock rate. The divider is zero-based, so 0 corresponds to divide-by-one. The PWM clock divider sets the minimum pulse width and thus the PWM period.

Example: mpwm_clkdiv[14:0] = 3

The minimum pulse width is one PWM clock period. Assume that the system clock is 40 MHz (25 nsec period). In this example the PWM clock rate divider is 3+1 = 4, and the minimum pulse width is thus 25 x 4 = 100 nsec. The PWM period is the minimum pulse width x 256. So in our example, the PWM period is 25.6 uS, for a PWM frequency of ~39KHz.

mpwmReload0 - mpwmReload2 (0xFCC1, 0xFCC5, 0xFCC9)

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
polarity	phase							mpwm_pulse							

cleared to zero on reset.

Bit [15] 0: The even-numbered P1 port pin is driven by the pulse generator (*forward* direction).
1: The odd-numbered P1 port pin is driven by the pulse generator (*reverse* direction).

Bit [14] 0: Pulses are active high. The inactive P1 port pin is driven low.
1: Pulses are active low. The inactive P1 port pin is driven high.

Bits [13:8] Reserved. These bits should be written with zeroes.

Bits [7:0] PWM pulse width, 0 to 255.

If the pulse width is zero (either polarity), the PWM count and reload functions are halted for this unit. Thus, zeroing this field will disable the corresponding PWM unit. In this state both PWM outputs will be in their inactive state as determined by the phase bit, bit [14].

mpwmLink (0xFCD0)

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
	all_stop														

cleared to zero on reset.

This register provides a means to enable an emergency stop function.

Emergency Stop

Emergency stop exists as an option to quickly disable all PWM activity when an external hardware fault condition is sensed. The fault condition must pull port input P1.1 LOW. For example, an overshoot limit switch connected to P1.1 might be used to detect the fault.

If mpwmLink[14] is set and if port P1.1 input is pulled LOW, all pwmReload*n*[7:0] register values will be cleared, each mpwmCtrl*n*[15] will be cleared, all PWM counter activity will stop, and all PWM output ports will be set to their inactive state as determined by each unit's phase bit (mpwmReload*n*[14]).

Both mpwmReload*n*[15:14] bits are unchanged. The application can attempt a recovery by clearing mpwmLink[14]. This will disable the fault detector. The application can then try to move a motor (or motors) while polling P1.1 to test if the fault clears.

Sensor Section

The sensor section has three sensor units that measure travel distance by counting state transitions that are typically generated by IR interrupter tachometers sensors, which in turn are linked to the mechanical system.

Each of the three sensor units can be configured for either quadrature or single-ended input signals. Quadrature sensor arrays provide 2 pulse trains, with a 90 degree phase shift between the 2 trains. This provides a simple means of determining direction of travel. Single-ended arrays only provide one pulse train, and can be used where direction information is not required.

The sensor section, along with the PWM section described above, provide the application with the controls needed to accomplish software-based closed-loop motion control. The application can also measure sensor rate by sampling the sensor counts at regular intervals.

Each sensor array is assigned two I/O pins in quadrature mode, and one I/O pin in single-ended mode. The pin(s) must be configured as inputs (rather than outputs) by the application. Configuring the input pull-up type (weak, strong, Hi-Z) will depend on the sensors used, and is the responsibility of the application developer. The sensor assigned GPIO pins are Schmitt Trigger inputs to improve noise immunity

The sensor block does not have any low-pass, signal conditioning, or glitch prevention hardware other than the quadrature mode's inherent grey-scale noise immunity and the hysteresis in the GPIO inputs. This is because the system has a sensor rate range from "DC" to NLP-5x's system clock/4 (or nominally 10MHz sensor toggle rate). It will be the application's responsibility to design any signal conditioning for the expected sensor rate.

Quadrature Mode Operation

Sensor I/O Pin Assignment:

Sensor Unit #0	P0[9:8]
Sensor Unit #1	P0[11:10]
Sensor Unit #2	P0[13:12]

Note: P0[15:8] are Schmitt Trigger Inputs.

In quadrature mode, the motor direction is determined by which direction the 2-bit grey scale value (of the quadrature inputs) changes, as follows:

1. 00->01->11->10->00 is defined as *forward*.
2. 00->10->11->01->00 is defined as *reverse*.

These are the only valid sequences. Sequences other than the above are illegal, and will be ignored as noise-created glitches.

A *forward* step increments the sensorUpCnt counter and *clears* the direction bit (sens*n*dir) in sensorCtrl.
A *reverse* step increments the sensorDnCnt counter and *sets* the direction bit (sens*n*dir) in sensorCtrl.

Each legal transition between states increments the appropriate counter. Thus the count will increment 4 times faster than an actual sensor rate.

Single-Ended Operation

Sensor I/O Pin Assignment:

Sensor Unit #0 P0[8]
Sensor Unit #1 P0[10]
Sensor Unit #2 P0[12]

When used with single-ended sensors, only the even numbered I/O pin of the pair is used. To enable this mode, the sens*n*sngl bit must be set in sensorCtrl. This mode allows use of the odd I/O pin as a general purpose I/O pin. Also, only the sensorUpCnt*n* counter is used in single-ended mode.

In single-ended mode, the sensorUpCnt register is incremented on the falling edge of the associated sensor. Since we only count on one edge in single mode, the smallest resolution distance (in distance/measured step) is four times as large as that of quadrature mode

sensorCtrl (0xFCD1)

15	14	13	12	11	10	09	08
		sens2clr	sens2sngl	sens1clr	sens1sngl	sens0clr	sens0sngl
07	06	05	04	03	02	01	00
		sens2dir	sens2ovr	sens1dir	sens1ovr	sens0dir	sens0ovr

Cleared to zero on reset.

Bits [13,11,9] Write-only.
1: The associated up and down counters and sens*n*ovr bit are cleared to zero.
0: No function.

Bits [12, 10, 8] Read/Write.
1: Single-ended mode.
0: Quadrature mode.

Bits [5, 3, 1] Read Only. These bits are meaningful only in quadrature mode. Each returns the direction determined from the 2 most recent valid samples:
0: forward
1: reverse.

Bits [4, 2, 0] Read Only. These bits are set when either of the associated sensorUpCnt or sensorDnCnt counters overflows. It is cleared by writing a “1” to the associated sens*n*clr bit.

sensorUpCnt0 – sensorUpCnt2 (0xFCC2, 0xFCC6, 0xFCCA)

15	14	13	12	11	10	09	08
msb							
07	06	05	04	03	02	01	00
							lsb

cleared to zero on reset. These are read/only registers.

sensorDnCnt0 – sensorDnCnt2 (0xFCC3, 0xFCC7, 0xFCCB)

15	14	13	12	11	10	09	08
msb							
07	06	05	04	03	02	01	00
							lsb

cleared to zero on reset. These are read/only registers.

APPENDIX A - Instruction Set Summary

Move Instructions

Name	Syntax	Description
MOV	mov rX, rY	rX ←rY
MOV	mov cX, rY	cX ←rY
MOV	mov rX, cY	rX ← cY
MOV	mov rX, IMM4S	rX ← IMM4S
MOVL	movl rX, IMM8U	rX[7:0] ← IMM8U
MOVH	movh rX, IMM8U	rX[15:8] ← IMM8U
MOVL	movl cX, IMM8U	cX[7:0] ← IMM8U cX = {%fmode, %loop0, %loop1, %loop2, %loop3, %guard}
MOVH	movh cX, IMM8U	cX[15:8] ← IMM8U cX = {%fmode, %loop0, %loop1, %loop2, %loop3, %guard}

Synthetic Move Instructions

Name	Syntax	Description

LDA	lda rX, LABEL	Replaced by:
		movl rX, LABEL[7:0]
		movh rX, LABEL[15:8]

MOV	mov rX, IMM	if -8 <= IMM <= 7, then replaced by:
		mov rX, IMM4S
		else replaced by:
		movl rX, IMM[7:0]
		movh rX, IMM[15:8]

MOV	mov cX, IMM16U	cX = {%fmode, %loop0,%loop1, %loop2, %loop3, %guard}
		Replaced by:
		movl cX, IMM16U[7:0]
		movh cX, IMM16U[15:8]

Load Register(s) from Memory Instructions

Name	Syntax	Description
LD	ld rX, rY [,n]	rX ← mem[rY+n], -4 <= n <= 3
LDU	ldu rX, rY, n	n = {-2, -1, 1, 2} rX ← mem[rY] rY += n
LDDU	lddu rX.e, rY, 2	rX ← mem[rY] r(X+1) ← mem[rY+1] rY += 2
LDDU	lddu rX.e, rY, -2	rX ← mem[rY-1] r(X+1) ← mem[rY] rY += -2
LDX	ldx rX, rY.e	rX ← mem[rY + r(Y+1)]
LDXU	ldxu rX, rY.e	rX ← mem[rY + r(Y+1)] rY += r(Y+1)

Store Register(s) to Memory Instructions

Name	Syntax	Description
ST	st rX, rY [,n]	mem[rY+n] ←rX, -4 <= n <= 3
STU	st rX, rY, n	n = {-2, -1, 1, 2} mem[rY] ← rX rY += n
STDU	stdu rX.e, rY, 2	mem[rY] ← rX mem[rY+1] ← r(X+1) rY += 2
STDU	stdu rX.e, rY, -2	mem[rY-1] ← rX mem[rY] ← r(X+1) rY += -2
STX	stx rX, rY.e	mem[rY + r(Y+1)] ← rX
STXU	stxu rX, rY.e	mem[rY + r(Y+1)] ← rX rY += r(Y+1)

Branch Instructions

Unconditional branch instructions use a 12-bit signed offset from the current program counter to branch within the range of -2048 to +2047 words. Conditional branch instructions use an 8-bit signed offset from the current program counter to branch within the range -128 to +127 words. The agn0, agn1, agn2, and agn3 instructions use an 8-bit negative offset in the range -256 to -1 words. The call instruction uses a 13-bit signed offset that has the LSB always zero, to branch within a range -4096 to +4094 of the current program counter.

Name	Syntax	Description
BR	br rX	%pc ←rX
BR	br LABEL	%pc ←LABEL
BZ	bz LABEL	if hwf<z> %pc ← LABEL else %pc ← %pc + 1
BNZ	bnz LABEL	if !hwf<z> %pc ← LABEL else %pc ← %pc + 1
BLT	blt LABEL	if !hwf<ge> %pc ← LABEL else %pc ← %pc + 1
BLE	ble LABEL	if !hwf<gt> %pc ← LABEL else %pc ← %pc + 1
BGT	bgt LABEL	if hwf<gt> %pc ← LABEL else %pc ← %pc + 1
BGE	bge LABEL	if hwf<ge> %pc ← LABEL else %pc ← %pc + 1
BOV	bov LABEL	if hwf<v> %pc ← LABEL else %pc ← %pc + 1
BNOV	bnov LABEL	if !hwf<v> %pc ← LABEL else %pc ← %pc + 1
BC	bc LABEL	if hwf<c> %pc ← LABEL else %pc ← %pc + 1

Branch Instructions (Continued)

Name	Syntax	Description
BNC	bnc LABEL	if !hwf<c> %pc ← LABEL else %pc ← %pc + 1

AGN0	agn0 LABEL	if (%loop0 != 0) {
		%pc ← LABEL
		%loop0 ← %loop0 – 1
		}
		else {
		%pc ← %pc + 1
		%loop0 ← %loop0 – 1
		}

AGN1	agn1 LABEL	if (%loop1 != 0) {
		%pc ← LABEL
		%loop1 ← %loop1 – 1
		}
		else {
		%pc ← %pc + 1
		%loop1 ← %loop1 – 1
		}

AGN2	agn2 LABEL	if (%loop2 != 0) {
		%pc ← LABEL
		%loop2 ← %loop2 – 1
		}
		else {
		%pc ← %pc + 1
		%loop2 ← %loop2 – 1
		}

AGN3	agn3 LABEL	if (%loop3 != 0) {
		%pc ← LABEL
		%loop3 ← %loop3 – 1
		}
		else {
		%pc ← %pc + 1
		%loop3 ← %loop3 – 1
		}

CALL	call rX	%rpc ← %pc + 1 %pc ← rX
CALL *1	call LABEL	%rpc ← %pc + 1 %pc ← LABEL
RET	ret	%pc ← %rpc
RETI	reti	%pc ← %tpc %imask<gie> ← %imask<pgie> %ip0<epl> ← %ip0k<pepl>

Note *1: The assembler may insert a NOP instruction prior to a relative CALL instruction in order to align the CALL instruction to be at an even address.

Shift Instructions

SHLL and SHLA both shift left and insert zeros in the least significant bits. SHLA will honor the SAT and SRA bits in %fmode, while SHLL will not. SHRA shifts right with sign extension, SHRL shifts right inserting zeros in the most significant bits.

These instructions clear the carry flag unconditionally.

Name	Syntax	Description
REVB	revb rX, IMM4U	Reverses order of rX[IMM4U:0]. If IMM4U < 15, rx[15: (IMM4U+1)] is set to zero.
SHLA	shla rX, IMM4U	rX ← rX << IMM4U
SHLA	shla rX, rY	rX ← rX << rY[3:0]
SHLA.E	shla.e rX.e, IMM5U	{r(X+1) rX} ← {r(X+1) rX} << IMM5U
SHLA.E	shla.e rX.e, rY	{r(X+1) rX} ← {r(X+1) rX} << rY[4:0]
SHLL	shll rX, IMM4U	rX ← rX << IMM4U
SHLL	shll rX, rY	rX ← rX << rY[3:0]
SHLL.E	shll,e rX.e, IMM5U	{r(X+1) rX} ← {r(X+1) rX} << IMM5U
SHLL.E	shll.e rX.e, rY	{r(X+1) rX} ← {r(X+1) rX} << rY[4:0]
SHRA	shra rX, IMM4U	rX ← rX >> IMM4U (sign extend)
SHRA	shra rX, rY	rX ← rX >> rY[3:0] (sign extend)
SHRA.E	shra.e rX.e, IMM5U	{r(X+1) rX} ← {r(X+1) rX} >> IMM5U (sign extend)
SHRA.E	shra.e rX.e, rY	{r(X+1) rX} ← {r(X+1) rX} >> rY[4:0] (sign extend)
SHRL	shrl rX, IMM4U	rX ← rX >> IMM4U
SHRL	shrl rX, rY	rX ← rX >> rY[3:0]
SHRL.E	shrl.e rX.e, IMM5U	{r(X+1) rX} ← {r(X+1) rX} >> IMM5U
SHRL.E	shrl.e rX.e, rY	{r(X+1) rX} ← {r(X+1) rX} >> rY[4:0]

Bit Manipulation Instructions

cX in the table below refers to one of the following control registers: %fmode, %tc, %imask, %ip0, %ip1, %guard, %hwflag, %ireq, %vitr, %smode, %amode.

Name	Syntax	Description
BITC	bitc rX, IMM4U	rX &= ~(1 << IMM4U)
BITC	bitc cX, IMM4U	cX &= ~(1 << IMM4U)
BITI	biti rX, IMM4U	rX ^= (1 << IMM4U)
BITI	biti cX, IMM4U	cX ^= (1 << IMM4U)
BITS	bits rX, IMM4U	rX |= (1 << IMM4U)
BITS	bits cX, IMM4U	cX |= (1 << IMM4U)
BITT	bitt rX, IMM4U	hwf<z> ← !(rX & (1 << IMM4U))
BITT	bitt cX, IMM4U	hwf<z> ← !(cX & (1 << IMM4U))

Logical Instructions

Name	Syntax	Description
AND	and rX, rY	rX &= rY
AND.E	and.e rX.e, rY.e	{r(X+1) rX} &= {r(Y+1) rY}
NOT	not rX, rY	rX ← ~rY
NOT.E	not.e rX.e, rY.e	{r(X+1) rX} ← ~{r(Y+1) rY}
OR	or rX, rY	rX |= rY
OR.E	or.e rX.e, rY.e	{r(X+1) rX} |= {r(Y+1) rY}
XOR	xor rX, rY	rX ^= rY
XOR.E	xor.e rX.e, rY.e	{r(X+1) rX} ^= {r(Y+1) rY}

Arithmetic Instructions

Name	Syntax	Description
ABS	abs rX, rY	rX ← ABS(rX, rY)
ABS.E	abs.e rX.e, rY.e	{r(X+1) rX} ← ABS({r(Y+1) rY})
ADD	add rX, rY	rX += rY
ADD	add rX, IMM4S	rX += IMM4S
ADD.E	add.e rX.e, rY.e	{r(X+1) rX} += {r(Y+1) rY}
ADDC.E	addc.e rX.e, rY.e	{r(X+1) rX} += {r(Y+1) rY} + carry
CMP	cmp rX, rY	hwf<ge> set if rX >= rY (signed comparison) hwf<gt> set if rX > rY (signed comparison) hwf<c> set if rX >= rY (unsigned comparison) hwf<z> set if rX == rY
CMP.E	cmp.e rX.e, rY.e	hwf<ge> set if {r(x+1) rX} >= {r(Y+1) rY} (signed comparison) hwf<gt> set if {r(x+1) rX} > {r(Y+1) rY} (signed comparison) hwf<c> set if {r(x+1) rX} >= {r(Y+1) rY} (unsigned comparison) hwf<z> set if {r(x+1) rX} == {r(Y+1) rY}
MIN	min rX, rY	rX ← MIN(rX, rY) (signed comparison)
MIN.E	min.e rX.e, rY.e	{r(X+1) rX} ← MIN({r(X+1) rX}, {r(Y+1) rY}) (signed comparison)
MAX	max rX, rY	rX ← MAXrX, rY) (signed comparison)
MAX.E	max.e rX.e, rY.e	{r(X+1) rX} ← MAX({r(X+1) rX}, {r(Y+1) rY}) (signed comparison)
NEG	neg rX, rY	rX ← -rY
NEG.E	neg.e rX.e, rY.e	{r(X+1) rX} ← -{r(Y+1) rY}
NORM	norm rX, rY	If rY == 0 then rX = 0 else if rY == -1 then rX = 15 else if rY >= 0 then rX = 14 - bit position of leading 1 in rY else rX = 14 - bit position of leading 0 in rY
NORM.E	norm.e rX, rY.e	If rY.e == 0 then rX = 0 else if rY.e == -1 then rX = 31 else if rY.e >= 0 then rX = 30 - bit position of leading 1 in rY.e else rX = 30 - bit position of leading 0 in rY

Arithmetic Instructions (Continued)

Name	Syntax	Description
PADD.A	padd.a rX.e, rY.e	r0 ← rX + rY r1 ← r(X+1) + r(Y+1)
PADD.B	padd.b rX.e, rY.e	r2 ← rX + rY r3 ← r(X+1) + r(Y+1)
PSUB.A	psub.a rX.e, rY.e	r0 ← rX – rY r1 ← r(X+1) - r(Y+1)
PSUB.B	psub.b rX.e, rY.e	r2 ← rX – rY r3 ← r(X+1) - r(Y+1)
ROUND.E	round.e rX.e, rY.e	{r(X+1) rX} ← {r(Y+1) rY} + 0x00008000
SUB	sub rX, rY	rX ← rX – rY
SUB.E	addc.e rX.e, rY.e	{r(X+1), rX} ← {r(X+1), rX} - r(Y+1), rY}
SUBC.E	cmp rX, rY	{r(X+1), rX} ← {r(X+1), rX} - r(Y+1), rY} – (~hwf<c>)

Multiplication Instructions

Note: all of these instructions except IMUl.A and IMUL.B honor the %fmode flags SAT, Q15, and MRE.

Name	Syntax	Description
CMULI.A	cmuli.a rX, rY	{r1 r0} ← rX * r(Y + 1) + r(X + 1) * rY
CMULI.B	cmuli.b rX, rY	{r3 r2} ← rX * r(Y + 1) + r(X + 1) * rY
CMULR.A	cmulr.a rX, rY	{r1 r0} ← r(X + 1) * r(Y + 1) - rX * rY
CMULR.B	cmulr.b rX, rY	{r3 r2} ← r(X + 1) * r(Y + 1) - rX * rY
DMUL.A	dmul.a rX, rY	{r1 r0} ← ({r(X+1) rX} * {r(Y+1) rY)}) >> 32
DMUL.B	dmul.b rX, rY	{r3 r2} ← ({r(X+1) rX} * {r(Y+1) rY)}) >> 32
IMUL.A	imul.a rX, rY	{r1 r0} ← rX * rY (%fmode flags ignored)
IMUL.B	imul.b rX, rY	{r3 r2} ← rX * rY (%fmode flags ignored)
MUL.A	mul.a rX, rY	{r1 r0} ← rX * rY
MUL.B	mul.b rX, rY	{r3 r2} ← rX * rY
MULN.A	muln.a rX, rY	{r1 r0} ← -rX * rY
MULN.B	muln.b rX, rY	{r3 r2} ← -rX * rY

Multiply-Accumulate Instructions

Name	Syntax	Description
CMACI.A	cmaci.a rX, rY	{g0 r1 r0} += rX * r(Y + 1) + r(X + 1) * rY
CMACI.B	cmaci.b rX, rY	{g1 r3 r2} += rX * r(Y + 1) + r(X + 1) * rY
CMACR.A	cmacr.a rX, rY	{g0 r1 r0} += r(X + 1) * r(Y + 1) - rX * rY
CMACR.B	cmacr.b rX, rY	{g1 r3 r2} += r(X + 1) * r(Y + 1) - rX * rY
DMAC.A	dmac.a rX, rY	{g0 r1 r0} += ({r(X+1) rX} * {r(Y+1) rY)}) >> 32
DMAC.B	dmac.b rX, rY	{g1 r3 r2} += ({r(X+1) rX} * {r(Y+1) rY)}) >> 32
MAC.A	mac.a rX, rY	{g0 r1 r0} += rX * rY
MAC.B	mac.b rX, rY	{g1 r3 r2} += rX * rY
MAC2.A	mac2.a rX, rY	{g0 r1 r0} += rX * rY + r(X+1) * r(Y+1)
MAC2.B	mac2.b rX, rY	{g1 r3 r2} += rX * rY + r(X+1) * r(Y+1)
MACN.A	macn.a rX, rY	{r1 r0} += -rX * rY
MACN.B	macn.b rX, rY	{r3 r2} += -rX * rY

APPENDIX B - Peripheral Registers Summary

Address	Name	Functions	Page
0xFC00	p0Out	P0[15:0] outputs	58
0xFC01	p0In	P0[15:0] inputs	58
0xFC02	p0CtrlA	P0 configuration	58
0xFC03	p0CtrlB	P0 configuration	58
0xFC04	p1Out	P1[15:0] outputs	58
0xFC05	p1In	P1[15:0] inputs	58
0xFC06	p1CtrlA	P1 configuration	58
0xFC07	p1CtrlB	P1 configuration	58
0xFC08	t2Reload	timer2 rate	71
0xFC09	t2Value	timer2 current count	71
0xFC0A	timerCtrl	timers #2 and 3 control	73
0xFC0B	t3Reload	timer3 rate	73
0xFC0C	t3Value	timer3 current count	73
0xFC10	cmpACtrl	comparator A control and status bits	94
0xFC11	cmpBCtrl	comparator B control and status bits	95
0xFC12	pgaCtrl1	preamplifier #1 gain, power	78
0xFC13	pgaCtrl2	preamplifier #2 gain, power	78
0xFC14	lineCtrl	line level input channel control	79
0xFC20	adcCtrl1	ADC control #1	81
0xFC21	adcCtrl2	ADC control #1	82
0xFC22	adcSR	ADC sample rate	83
0xFC23	adcCtrl3	ADC clock delay	82
0xFC24	adcFifo3	ADC FIFO read slot 3	86
0xFC25	adcFifo2	ADC FIFO read slot 2	87
0xFC26	adcFifo1	ADC FIFO read slot 1	87
0xFC27	adcStatus	ADC FIFO status	87
0xFC28	pwmCtrl	PWM control	89
0xFC29	pwmFrame	PWM frame size	90
0xFC2A	pwmFifo	PWM FIFO write	90

Address	Name	Functions	Page
0xFC2B	pwmStatus	PWM FIFO status	91
0xFC2C	dacCtrl	DAC control	96
0xFC2D	dacFifo2	DAC FIFO write channel 2	97
0xFC2E	dacFifo1	DAC FIFO write channel 1	97
0xFC2F	dacStatus	DAC FIFO status	97
0xFC30	miMask0	merged interrupt mask register 0	53
0xFC31	miMask1	merged interrupt mask register 1	53
0xFC32	miMask2	merged interrupt mask register 2	53
0xFC33	miMask3	merged interrupt mask register 3	53
0xFC34	miMask4	merged interrupt mask register 4	53
0xFC35	miStatus	merged interrupt state (R) acknowledge merged interrupt (W)	53
0xFC38	p2Out	P2[7:0] outputs	58
0xFC39	p2In	P2[7:0] inputs	58
0xFC3A	p2CtrlA	P2 configuration	58
0xFC3B	p2CtrlB	P2 configuration	58
0xFC40	lcdCtrl	enable, configuration, timing	122
0xFC42	lcdP0Assign	specify P0 IO bits used for LCD	124
0xFC43	lcdP1Assign	specify P1 IO bits used for LCD	124
0xFC44	lcdData0	pixel bits, COM0, SEGS0-15	126
0xFC45	lcdData1	pixel bits, COM0, SEGS16-31	126
0xFC46	lcdData2	pixel bits, COM1, SEGS0-15	126
0xFC47	lcdData3	pixel bits, COM1, SEGS16-31	126
0xFC48	lcdData4	pixel bits, COM2, SEGS0-15	126
0xFC49	lcdData5	pixel bits, COM2, SEGS16-31	126
0xFC4A	lcdData6	pixel bits, COM3, SEGS0-15	126
0xFC4B	lcdData7	pixel bits, COM3, SEGS16-31	126
0xFC50	irConfig	enable, mode, carrier frequency	114
0xFC51	irStatus	receive state, bit value	115
0xFC52	irXmtCntLow	duration of transmit low pulse	115
0xFC53	irXmtCntHigh	duration of transmit high pulse	115
0xFC54	irRcvCntLow	duration of received low pulse	116

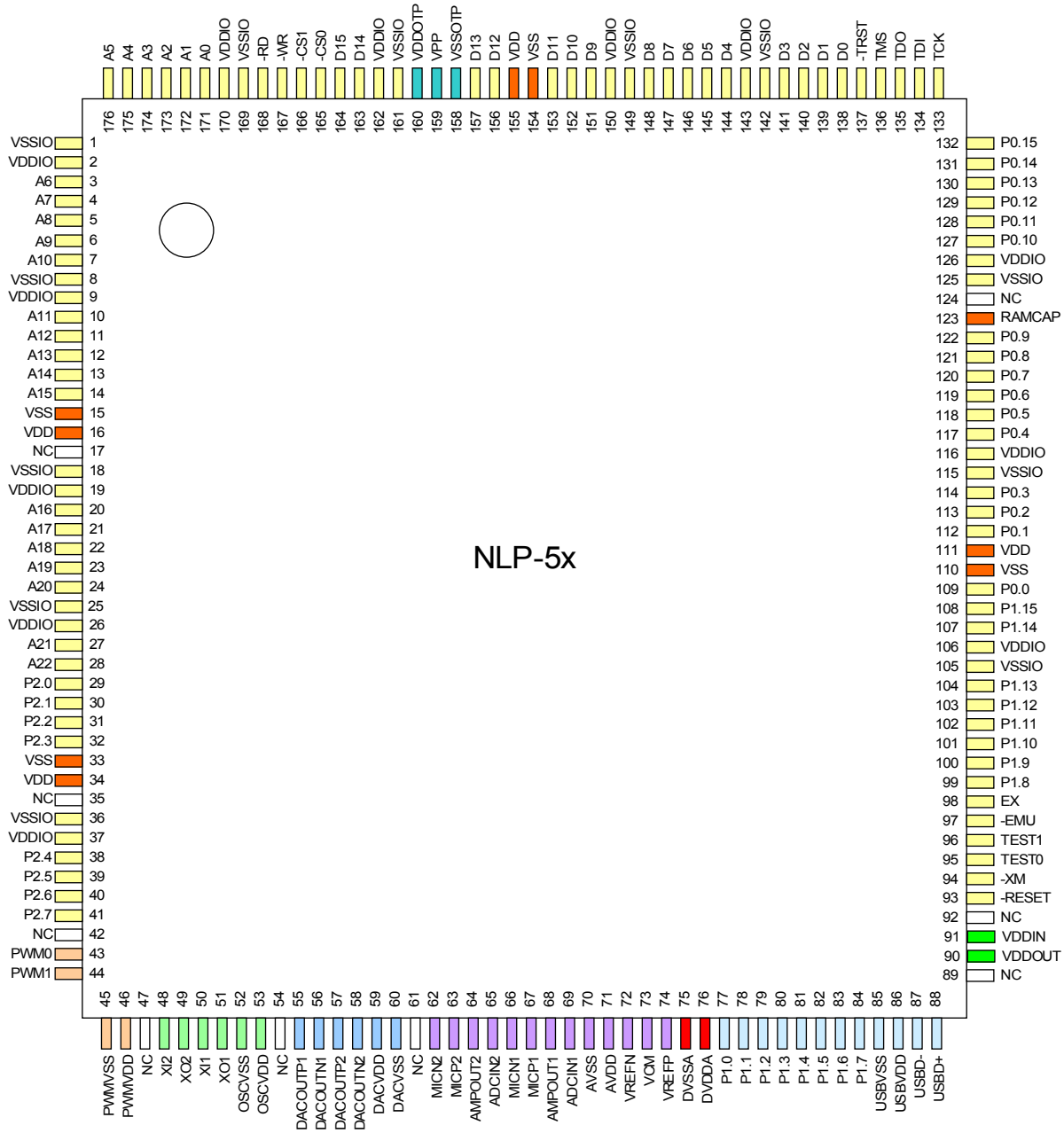
Address	Name	Functions	Page
0xFC55	irRcvCntHigh	duration of received high pulse	116
0xFC5A	usbCtrl	USB control register	119
0xFC5B	usbCmd	USB transfer command	120
0xFC5C	usbData0		118
0xFC5D	usbData1		118
0xFC5E	usbData2		118
0xFC5F	usbData3		118
0xFC60	lsoCtrl	low speed oscillators control	34
0xFC61	hsoCtrl	high speed oscillator / PLL control	33
0xFC62	clkSelect	cpu, usb clock selects	37
0xFC63	wakeMask	sleep mode wakeup event mask	41
0xFC64	sleep	enter sleep mode (W), wake up event (R)	40
0xFC65	iowEvent	I/O wakeup event register	42
0xFC66	iowMask	I/O wakeup mask register	42
0xFC67	iowPolarity	I/O wakeup polarity register	43
0xFC68	iowLevel	I/O wakeup level/edge register	43
0xFC69	miscCtrl	miscellaneous interrupt, system control	43
0xFC6A	miscStatus	miscellaneous interrupt, system status	45
0xFC6B	hpiData	host/peripheral interface data register	46

Address	Name	Functions	Page
0xFC70	sspData0	synchronous serial port data #0	108
0xFC71	sspData1	synchronous serial port data #1	108
0xFC72	sspData2	synchronous serial port data #2	108
0xFC73	sspData3	synchronous serial port data #3	108
0xFC74	sspCtrl	synchronous serial port control	108
0xFC75	sspStatus	synchronous serial port status	111
0xFC78	uartCtrl	UART control	100
0xFC79	uartBaud	UART baud rate selection	99
0xFC7A	uartData	UART data register	101
0xFC84	xmConfig	external memory configuration	25
0xFC85	dSeg0	data segment 0 address	23
0xFC86	dSeg1	data segment 1 address	23
0xFC87	cBank	code bank register	20
0xFC88	dBound	boundary between data segments	23
0xFC8A	bpAdrs	breakpoint address	21
0xFC8B	bpBank	breakpoint bank	21
0xFC8C	bpData	breakpoint data	21
0xFC91	testMode	test mode register	Error: Reference source not found
0xFCA0	xp0Out	aux port 0 (D[15:0]) output	60
0xFCA1	xp0In	aux port 0 (D[15:0]) input	60
0xFCA2	xp0CtrlA	aux port 0 (D[15:0]) configuration	60
0xFCA3	xp0CtrlB	aux port 0 (D[15:0]) configuration	60

Address	Name	Functions	Page
0xFCA4	xp1Out	aux port 1 (A[15:0]) output	60
0xFCA5	xp1In	aux port 1 (A[15:0]) input	60
0xFCA6	xp1CtrlA	aux port 1 (A[15:0]) configuration	60
0xFCA7	xp1CtrlB	aux port 1 (A[15:0]) configuration	60
0xFCA8	xp2Out	aux port 2 (A[22:16]) output	60
0xFCA9	xp2In	aux port 2 (A[22:16]) input	60
0xFCAA	xp2CtrlA	aux port 2 (A[22:16]) configuration	60
0xFCAB	xp2CtrlB	aux port 2 (A[22:16]) configuration	60
0xFCC0	mpwmCtrl0	motor PWM #0 control	129
0xFCC1	mpwmReload0	motor PWM #0 reload value	130
0xFCC2	sensorUpCnt0	motor sensor #0 up counter	133
0xFCC3	sensorDnCnt0	motor sensor #0 down counter	133
0xFCC4	mpwmCtrl1	motor PWM #1 control	129
0xFCC5	mpwmReload1	motor PWM #1 reload value	130
0xFCC6	sensorUpCnt1	motor sensor #1 up counter	133
0xFCC7	sensorDnCnt1	motor sensor #1 down counter	133
0xFCC8	mpwmCtrl2	motor PWM #2 control	129
0xFCC9	mpwmReload2	motor PWM #2 reload value	130
0xFCCA	sensorUpCnt2	motor sensor #2 up counter	133
0xFCCB	sensorDnCnt2	motor sensor #2 down counter	133
0xFCD0	mpwmLink	motor PWM link register	130
0xFCD1	sensorCtrl	motor sensor control register	132

APPENDIX C – Package, Pin, and Pad Definitions

LQFP 176 Package Pin Assignments



Pin Functions

The NLP-5x die has 173 pads. It is available as tested die, tested wafer, or in a 176-pin LQFP package, of which 167 pins are used. 6 of the package pins are double bonded: PWMVDD, PWMVSS, PWM0, PWM1, VDDIN, and VDDOUT.

DIE Pad #		176 LQFP Pin #	Pin Name	Description	Type
1		1	VSSIO	Ground for I/O	GND
2		2	VDDIO	Power supply for I/O (1.62 to 3.60V)	PWR
3		3	A6	External memory address bus or general purpose I/O	I/O
4		4	A7	External memory address bus or general purpose I/O	I/O
5		5	A8	External memory address bus or general purpose I/O	I/O
6		6	A9	External memory address bus or general purpose I/O	I/O
7		7	A10	External memory address bus or general purpose I/O	I/O
8		8	VSSIO	Ground for I/O	GND
9		9	VDDIO	Power supply for I/O (1.62 to 3.60V)	PWR
10		10	A11	External memory address bus or general purpose I/O	I/O
11		11	A12	External memory address bus or general purpose I/O	I/O
12		12	A13	External memory address bus or general purpose I/O	I/O
13		13	A14	External memory address bus or general purpose I/O	I/O
14		14	A15	External memory address bus or general purpose I/O	I/O
15		15	VSS	Ground for digital core	GND
16		16	VDD	Power supply for digital core (1.8V +/- 10%)	PWR
		17	NC		
17		18	VSSIO	Ground for I/O	GND
18		19	VDDIO	Power supply for I/O (1.62 to 3.60V)	PWR
19		20	A16	External memory address bus or general purpose I/O	I/O
20		21	A17	External memory address bus or general purpose I/O	I/O
21		22	A18	External memory address bus or general purpose I/O	I/O
22		23	A19	External memory address bus or general purpose I/O	I/O
23		24	A20	External memory address bus or general purpose I/O	I/O
24		25	VSSIO	Ground for I/O	GND
25		26	VDDIO	Power supply for I/O (1.62 to 3.60V)	PWR
26		27	A21	External memory address bus or general purpose I/O	I/O
27		28	A22	External memory address bus or general purpose I/O	I/O
28		29	P2.0	General purpose I/O or SS_ (SSP function)	I/O
29		30	P2.1	General purpose I/O or SCLK (SSP function)	I/O
30		31	P2.2	General purpose I/O or MISO (SSP function)	I/O
31		32	P2.3	General purpose I/O or MOSI (SSP function)	I/O
32		33	VSS	Ground for digital core	GND
33		34	VDD	Power supply for digital core (1.8V +/- 10%)	PWR
		35	NC		

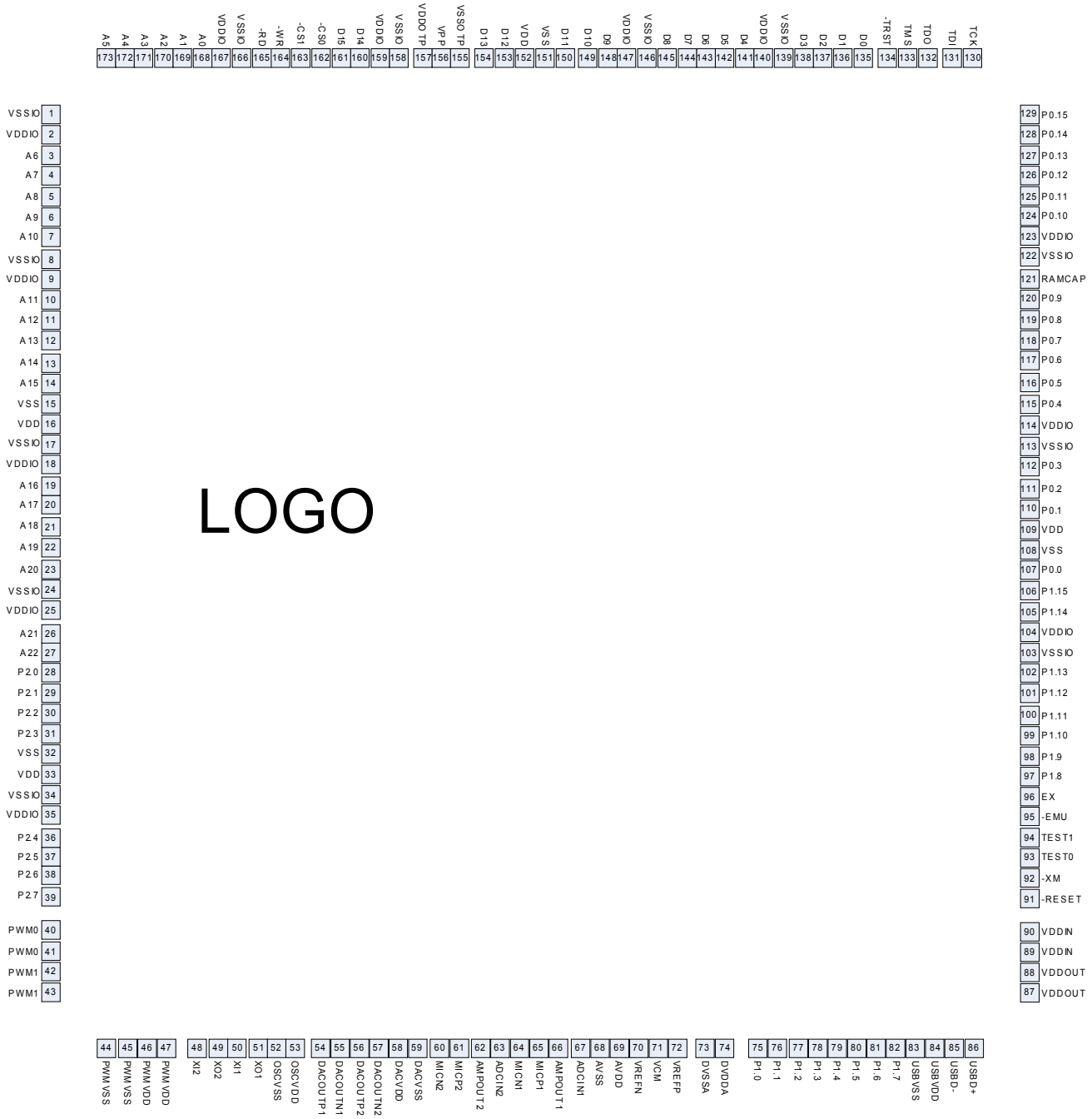
34		36	VSSIO	Ground for I/O	GND
35		37	VDDIO	Power supply for I/O (1.62 to 3.60V)	PWR
36		38	P2.4	General purpose I/O or RXIR (infrared receive)	I/O
37		39	P2.5	General purpose I/O or TXIR (infrared transmit)	I/O
38		40	P2.6	General purpose I/O or RXD (UART receive)	I/O
39		41	P2.7	General purpose I/O or TXD (UART transmit)	I/O
		42	NC		
40		43	PWM0	Pulse width modulator (PWM) output 0	O
41		43	PWM0	Pulse width modulator (PWM) output 0	O
42		44	PWM1	Pulse width modulator (PWM) output 1	O
43		44	PWM1	Pulse width modulator (PWM) output 1	O
44		45	PWMVSS	Ground for PWM output drivers	GND
45		45	PWMVSS	Ground for PWM output drivers	GND
46		46	PWMVDD	Power supply for PWM output drivers (1.62 to 3.60V)	PWR
47		46	PWMVDD	Power supply for PWM output drivers (1.62 to 3.60V)	PWR
		47	NC		
48		48	XI2	Crystal oscillator 2 input (32.768KHz)	I
49		49	XO2	Crystal oscillator 2 output	O
50		50	XI1	Crystal oscillator 1 input (2MHz to 8MHz)	I
51		51	XO1	Crystal oscillator 1 output	O
52		52	OSCVSS	Ground for crystal oscillators	GND
53		53	OSCVDD	Power supply for crystal oscillators (1.62 to 3.60V)	PWR
		54	NC		
54		55	DACOUTP1	DAC channel 1 positive output	O
55		56	DACOUTN1	DAC channel 1 negative output	O
56		57	DACOUTP2	DAC channel 2 positive output	O
57		58	DACOUTN2	DAC channel 2 negative output	O
58		59	DACVDD	Power supply for DAC output drivers (1.62 to 3.60V)	PWR
59		60	DACVSS	Ground for DAC output drivers	GND
		61	NC		
60		62	MICN2	Preamplifier channel 2 inverting input	I
61		63	MICP2	Preamplifier channel 2 non-inverting input	I
62		64	AMPOUT2	Preamplifier channel 2 output	O
63		65	ADCIN2	ADC channel 2 input	I
64		66	MICN1	Preamplifier channel 1 inverting input	I
65		67	MICP1	Preamplifier channel 1 non-inverting input	I
66		68	AMPOUT1	Preamplifier channel 1 output	O
67		69	ADCIN1	ADC channel 1 input	I
68		70	AVSS	Ground for analog circuits	GND
69		71	AVDD	Power supply for analog circuits (1.62 to 3.60V)	PWR
70		72	VREFN	ADC reference negative voltage	O
71		73	VCM	ADC common mode reference voltage	O
72		74	VREFP	ADC reference positive voltage	O
73		75	DVSSA	Ground for level shifters in analog circuits	GND

74		76	DVDDA	Power supply for level shifters in analog circuits (1.8V +/- 10%)	PWR
75		77	P1.0	General purpose I/O or analog inputs to comparators and third channel of ADC	I/O
76		78	P1.1	General purpose I/O or analog inputs to comparators and third channel of ADC	I/O
77		79	P1.2	General purpose I/O or analog inputs to comparators and third channel of ADC	I/O
78		80	P1.3	General purpose I/O or analog inputs to comparators and third channel of ADC	I/O
79		81	P1.4	General purpose I/O or analog inputs to comparators and third channel of ADC	I/O
80		82	P1.5	General purpose I/O or analog inputs to comparators and third channel of ADC	I/O
81		83	P1.6	General purpose I/O or analog inputs to comparators and third channel of ADC	I/O
82		84	P1.7	General purpose I/O or analog inputs to comparators and third channel of ADC	I/O
83		85	USBVSS	Ground for USB transceiver and P1.0 - P1.7	GND
84		86	USBVDD	Power for USB transceiver and P1.0 - P1.7 (1.62 to 3.60V, or 3.3V +/- 10% if USB is active)	PWR
85		87	USBD-	USB negative differential I/O	I/O
86		88	USBD+	USB positive differential I/O	I/O
		89	NC		
87		90	VDDOUT	On-chip regulator output (1.8V +/- 10%)	O
88		90	VDDOUT	On-chip regulator output (1.8V +/- 10%)	O
89		91	VDDIN	On-chip regulator input (2.0 to 3.6V)	I
90		91	VDDIN	On-chip regulator input (2.0 to 3.6V)	I
		92	NC		
91		93	-RESET	Reset input (active low)	I
92		94	-XM	External memory enable (active low)	I
93		95	TEST0	Test mode selection input 0	I
94		96	TEST1	Test mode selection input 1	I
95		97	-EMU	Emulation mode enable (active low)	I
96		98	EX	Emulation transmit flag output	O
97		99	P1.8	General purpose I/O	I/O
98		100	P1.9	General purpose I/O	I/O
99		101	P1.10	General purpose I/O	I/O
100		102	P1.11	General purpose I/O	I/O
101		103	P1.12	General purpose I/O	I/O
102		104	P1.13	General purpose I/O	I/O
103		105	VSSIO	Ground for I/O	GND
104		106	VDDIO	Power supply for I/O (1.62 to 3.60V)	PWR
105		107	P1.14	General purpose I/O	I/O
106		108	P1.15	General purpose I/O	I/O
107		109	P0.0	General purpose I/O	I/O
108		110	VSS	Ground for digital core	GND
109		111	VDD	Power supply for digital core (1.8V +/- 10%)	PWR

110		112	P0.1	General purpose I/O	I/O
111		113	P0.2	General purpose I/O	I/O
112		114	P0.3	General purpose I/O	I/O
113		115	VSSIO	Ground for I/O	GND
114		116	VDDIO	Power supply for I/O (1.62 to 3.60V)	PWR
115		117	P0.4	General purpose I/O	I/O
116		118	P0.5	General purpose I/O	I/O
117		119	P0.6	General purpose I/O	I/O
118		120	P0.7	General purpose I/O	I/O
119		121	P0.8	General purpose I/O with Schmitt trigger input buffer	I/O
120		122	P0.9	General purpose I/O with Schmitt trigger input buffer	I/O
121		123	RAMCAP	RAM digital power output (do not connect)	O
		124	NC		
122		125	VSSIO	Ground for I/O	GND
123		126	VDDIO	Power supply for I/O (1.62 to 3.60V)	PWR
124		127	P0.10	General purpose I/O with Schmitt trigger input buffer	I/O
125		128	P0.11	General purpose I/O with Schmitt trigger input buffer	I/O
126		129	P0.12	General purpose I/O with Schmitt trigger input buffer	I/O
127		130	P0.13	General purpose I/O with Schmitt trigger input buffer	I/O
128		131	P0.14	General purpose I/O with Schmitt trigger input buffer	I/O
129		132	P0.15	General purpose I/O with Schmitt trigger input buffer	I/O
130		133	TCK	JTAG serial clock in	I
131		134	TDI	JTAG serial data in	I
132		135	TDO	JTAG serial data out	O
133		136	TMS	JTAG serial mode select in	I
134		137	-TRST	JTAG serial interface reset (active low)	I
135		138	D0	External memory data bus or general purpose I/O	I/O
136		139	D1	External memory data bus or general purpose I/O	I/O
137		140	D2	External memory data bus or general purpose I/O	I/O
138		141	D3	External memory data bus or general purpose I/O	I/O
139		142	VSSIO	Ground for I/O	GND
140		143	VDDIO	Power supply for I/O (1.62 to 3.60V)	PWR
141		144	D4	External memory data bus or general purpose I/O	I/O
142		145	D5	External memory data bus or general purpose I/O	I/O
143		146	D6	External memory data bus or general purpose I/O	I/O
144		147	D7	External memory data bus or general purpose I/O	I/O
145		148	D8	External memory data bus or general purpose I/O	I/O
146		149	VSSIO	Ground for I/O	GND
147		150	VDDIO	Power supply for I/O (1.62 to 3.60V)	PWR
148		151	D9	External memory data bus or general purpose I/O	I/O
149		152	D10	External memory data bus or general purpose I/O	I/O
150		153	D11	External memory data bus or general purpose I/O	I/O
151		154	VSS	Ground for digital core	GND
152		155	VDD	Power supply for digital core (1.8V +/- 10%)	PWR
153		156	D12	External memory data bus or general purpose I/O	I/O
154		157	D13	External memory data bus or general purpose I/O	I/O
155		158	VSSOTP	Ground for OTP memories	GND

156		159	VPP	Programming power supply for OTP memories (1.8V +/- 10% for normal operation, 6.5V for programming)	PWR
157		160	VDDOTP	Power supply for OTP memories (1.8V +/- 10%)	PWR
158		161	VSSIO	Ground for I/O	GND
159		162	VDDIO	Power supply for I/O (1.62 to 3.60V)	PWR
160		163	D14	External memory data bus or general purpose I/O	I/O
161		164	D15	External memory data bus or general purpose I/O	I/O
162		165	-CS0	External memory chip select 0 (active low)	O
163		166	-CS1	External memory chip select 1 (active low)	O
164		167	-WR	External memory write strobe (active low)	O
165		168	-RD	External memory read strobe (active low)	O
166		169	VSSIO	Ground for I/O	GND
167		170	VDDIO	Power supply for I/O (1.62 to 3.60V)	PWR
168		171	A0	External memory address bus or general purpose I/O	I/O
169		172	A1	External memory address bus or general purpose I/O	I/O
170		173	A2	External memory address bus or general purpose I/O	I/O
171		174	A3	External memory address bus or general purpose I/O	I/O
172		175	A4	External memory address bus or general purpose I/O	I/O
173		176	A5	External memory address bus or general purpose I/O	I/O

Die Pad Ring

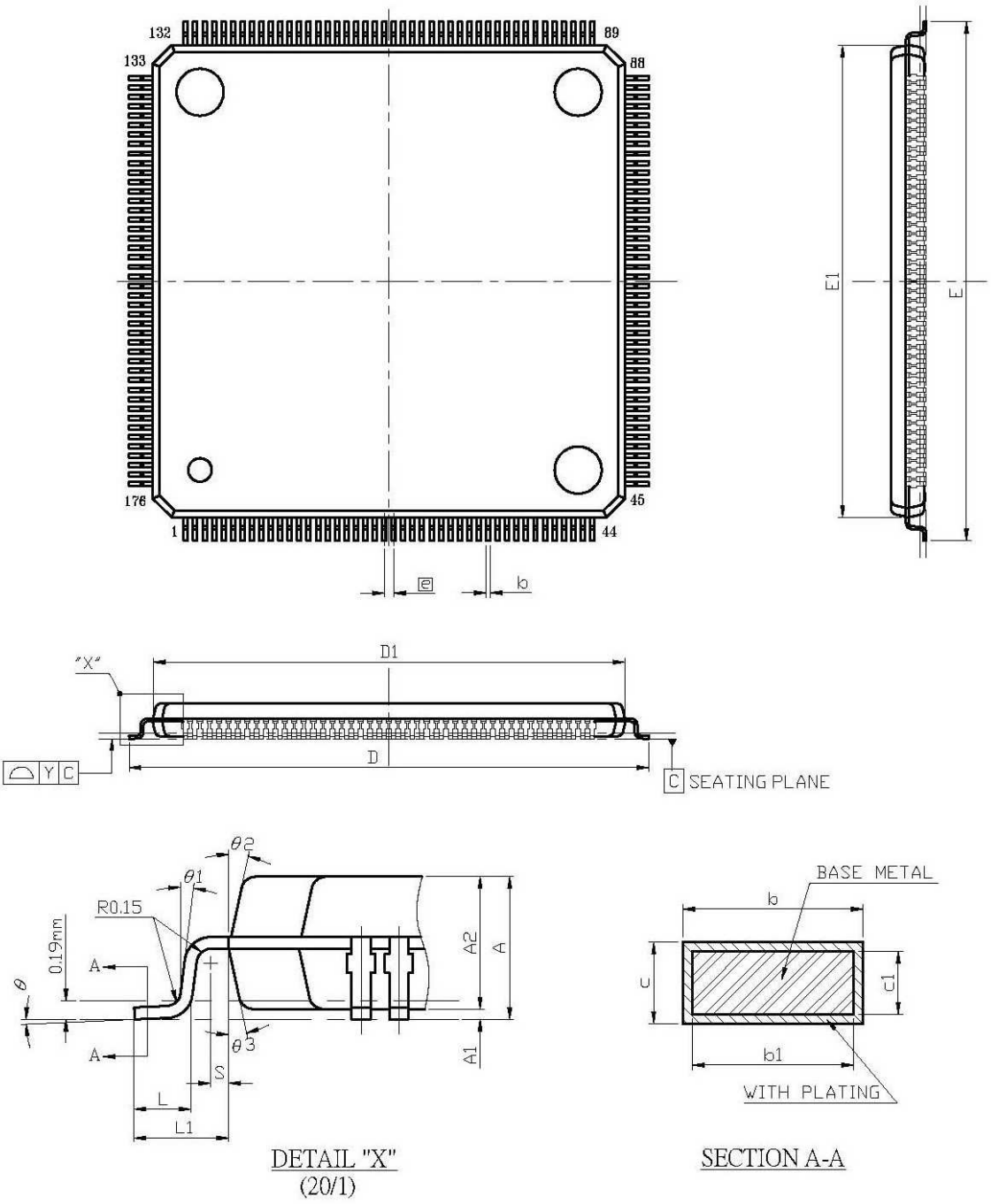


NLP-5x Die Bonding Pad Locations

DIE PAD #	PIN NAME	DIE X-um	DIE Y-um	DIE PAD #	PIN NAME	DIE X-um	DIE Y-um	DIE PAD #	PIN NAME	DIE X-um	DIE Y-um	DIE PAD #	PIN NAME	DIE X-um	DIE Y-um
1	VSSIO	99	3346	44	PWMVSS	371	99	87	VDDOUT	3724	371	130	TCK	3451	3618
2	VDDIO	99	3276	45	PWMVSS	441	99	88	VDDOUT	3724	441	131	TDI	3371	3618
3	A6	99	3206	46	PWMVDD	511	99	89	VDDIN	3724	511	132	TDO	3291	3618
4	A7	99	3136	47	PWMVDD	581	99	90	VDDIN	3724	581	133	TMS	3211	3618
5	A8	99	3066	48	XI2	681	99	91	-RESET	3724	686	134	-TRST	3131	3618
6	A9	99	2996	49	XO2	751	99	92	-XM	3724	756	135	D0	3051	3618
7	A10	99	2926	50	XI1	821	99	93	TEST0	3724	826	136	D1	2981	3618
8	VSSIO	99	2856	51	XO1	891	99	94	TEST1	3724	896	137	D2	2911	3618
9	VDDIO	99	2786	52	OSCVSS	961	99	95	-EMU	3724	966	138	D3	2841	3618
10	A11	99	2716	53	OSCVDD	1031	99	96	EX	3724	1036	139	VSSIO	2771	3618
11	A12	99	2646	54	DACOUTP1	1131	99	97	P1.8	3724	1106	140	VDDIO	2701	3618
12	A13	99	2576	55	DACOUTN1	1201	99	98	P1.9	3724	1176	141	D4	2631	3618
13	A14	99	2506	56	DACOUTP2	1271	99	99	P1.10	3724	1246	142	D5	2561	3618
14	A15	99	2436	57	DACOUTN2	1341	99	100	P1.11	3724	1316	143	D6	2491	3618
15	VSS	99	2366	58	DACVDD	1411	99	101	P1.12	3724	1386	144	D7	2421	3618
16	VDD	99	2296	59	DACVSS	1481	99	102	P1.13	3724	1456	145	D8	2351	3618
17	VSSIO	99	2226	60	MICN2	1561	99	103	VSSIO	3724	1526	146	VSSIO	2281	3618
18	VDDIO	99	2156	61	MICP2	1631	99	104	VDDIO	3724	1596	147	VDDIO	2211	3618
19	A16	99	2086	62	AMPOUT2	1701	99	105	P1.14	3724	1666	148	D9	2141	3618
20	A17	99	2016	63	ADCIN2	1771	99	106	P1.15	3724	1736	149	D10	2071	3618
21	A18	99	1946	64	MICN1	1841	99	107	P0.0	3724	1806	150	D11	2001	3618
22	A19	99	1876	65	MICP1	1911	99	108	VSS	3724	1876	151	VSS	1931	3618
23	A20	99	1806	66	AMPOUT1	1981	99	109	VDD	3724	1946	152	VDD	1861	3618
24	VSSIO	99	1736	67	ADCIN1	2051	99	110	P0.1	3724	2016	153	D12	1791	3618
25	VDDIO	99	1666	68	AVSS	2121	99	111	P0.2	3724	2086	154	D13	1721	3618
26	A21	99	1596	69	AVDD	2191	99	112	P0.3	3724	2156	155	VSSOTP	1641	3618
27	A22	99	1526	70	VREFN	2261	99	113	VSSIO	3724	2226	156	VPP	1571	3618
28	P2.0	99	1456	71	VCM	2331	99	114	VDDIO	3724	2296	157	VDDOTP	1501	3618
29	P2.1	99	1386	72	VREFP	2401	99	115	P0.4	3724	2366	158	VSSIO	1421	3618
30	P2.2	99	1316	73	DVSSA	2481	99	116	P0.5	3724	2436	159	VDDIO	1351	3618
31	P2.3	99	1246	74	DVDDA	2551	99	117	P0.6	3724	2506	160	D14	1281	3618
32	VSS	99	1176	75	P1.0	2651	99	118	P0.7	3724	2576	161	D15	1211	3618
33	VDD	99	1106	76	P1.1	2721	99	119	P0.8	3724	2646	162	-CS0	1141	3618
34	VSSIO	99	1036	77	P1.2	2791	99	120	P0.9	3724	2716	163	-CS1	1071	3618
35	VDDIO	99	966	78	P1.3	2861	99	121	RAMCAP	3724	2786	164	-WR	1001	3618
36	P2.4	99	896	79	P1.4	2931	99	122	VSSIO	3724	2856	165	-RD	931	3618
37	P2.5	99	826	80	P1.5	3001	99	123	VDDIO	3724	2926	166	VSSIO	861	3618
38	P2.6	99	756	81	P1.6	3071	99	124	P0.10	3724	2996	167	VDDIO	791	3618
39	P2.7	99	686	82	P1.7	3141	99	125	P0.11	3724	3066	168	A0	721	3618
40	PWM0	99	581	83	USBVSS	3211	99	126	P0.12	3724	3136	169	A1	651	3618
41	PWM0	99	511	84	USBVDD	3281	99	127	P0.13	3724	3206	170	A2	581	3618
42	PWM1	99	441	85	USBD-	3351	99	128	P0.14	3724	3276	171	A3	511	3618
43	PWM1	99	371	86	USBD+	3421	99	129	P0.15	3724	3346	172	A4	441	3618
												173	A5	371	3618

Notes:
1. Bonding pad opening is 60um x 60um
2. Coordinates are for center of bond pad opening, rounded to the nearest micron unit
3. Die size with seal ring and scribe line is approximately 3822um x 3717um
4. Coordinate origin is at bottom left corner of die

Mechanical Data



SYMBOL	DIMENSION (MM)			DIMENSION (MIL)		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	———	———	1.60	———	———	62.99
A1	0.05	———	0.15	1.97	———	5.91
A2	1.35	1.40	1.45	53.15	55.12	57.09
b	0.13	0.18	0.23	5.12	7.09	9.06
b1	0.13	0.16	0.19	5.12	6.30	7.48
c	0.09	———	0.20	3.54	———	7.87
c1	0.09	———	0.16	3.54	———	6.30
D	21.90	22.00	22.10	862.20	866.14	870.08
D1	19.90	20.00	20.10	783.46	787.40	791.34
E	21.90	22.00	22.10	862.20	866.14	870.08
E1	19.90	20.00	20.10	783.46	787.40	791.34
e	0.40 BSC			15.75 BSC		
L	0.45	0.60	0.75	17.72	23.62	29.53
L1	1.00 REF			39.37 REF		
R1	0.08	———	———	3.15	———	———
R2	0.08	———	0.20	3.15	———	7.87
Y	———	———	0.075	———	———	2.95
θ	0°	3.5°	7°	0°	3.5°	7°
θ1	0°	———	———	0°	———	———
θ2	11°	12°	13°	11°	12°	13°
θ3	11°	12°	13°	11°	12°	13°
S	0.20	———	———	7.87	———	———

NOTES:
1.REFER TO JEDEC MS-026/BFC REV. D
2.DIMENSION D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION.
ALLOWABLE PROTRUSION IS 0.25mm PER SIDE. D1 AND E1 ARE
MAXIMUM PLASTIC BODY SIZE DIMENSION INCLUDING MOLD MISMATCH.
3.DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE
DAMBAR PROTRUSION SHALL NOT CAUSE THE LEAD WIDTH TO EXCEED
THE MAXIMUM b DIMENSION BY MORE THAN 0.08mm.
4.ALL DIMENSIONS IN MILLIMETERS.
5.MODIFYIED MIL DIMENSION ,ADD TWO DECIMALS FOR CUSTOMER MANUFACTURE USE

APPENDIX D – Preliminary Electrical Characteristics

Notes:
1. Data in "Typ" column is at VDDIO=3.0V, TA=25C (unless otherwise stated)
2. Standard Operating Conditions: 1.62 V ≤ VDDIO ≤ 3.6 V, 0C ≤ TA ≤ 70C (unless otherwise stated)

DC Characteristics: Recommended Operating Conditions

Characteristic	Symbol	Min	Typ[1]	Max	Unit	Conditions[2]
Digital core voltage	VDD	1.62	1.80	1.98	V	
Digital voltage for LS	DVDDA	1.62	1.80	1.98	V	
OTP memory voltage	VDDOTP	1.62	1.80	1.98	V	
OTP program voltage	VPP					
Programming		6.25	6.5	6.75	V	
Reading		1.62	1.80	1.98	V	
I/O voltage	VDDIO	1.62	3.00	3.60	V	
Analog voltage	AVDD	1.62	3.00	3.60	V	
Crystal oscillator voltage	OSCVDD	1.62	3.00	3.60	V	
DAC I/O voltage	DACVDD	1.62	3.00	3.60	V	Note: DACVDD must be within AVDD +/- 0.3V.
PWM I/O voltage	PWMVDD	1.62	3.00	3.60	V	
USB and P1 I/O voltage	USBVDD					Note: USBVDD must be less than AVDD+0.3V
USB active		3.00	3.30	3.60	V	
USB not active		1.62	3.00	3.60	V	
On-chip regulator input voltage	VDDIN	2.00	3.00	3.60	V	
Ambient Temperature	TA	0	25	70	C	

DC Characteristics: Operating Current

Characteristic	Sym	Min	Typ[1]	Max	Unit	Conditions[2]
Supply Current, Active	IACT1		16		mA	9 MHz Non-Turbo Mode, PGAs disabled. Typical Application: SX synthesis (excludes speaker current).
Supply Current, Active	IACT2		35		mA	36 MHz Non-Turbo Mode, one PGA unit enabled. Typical Application: T2SI recognition.
Supply Current, Active	IACT3		36		mA	72 MHz Turbo Mode, PGAs disabled. Typical Applications: MP3 decoder, 24-voice MIDI synthesizer (excludes speaker current).
Supply Current, Idle	IIDLE		60		uA	High power regulation off, OSC2 enabled.
Supply Current, Sleep	ISLP		50		uA	High power regulation off, OSC2 disabled.

DC Characteristics: Digital I/O

Characteristic	Sym	Min	Typ[1]	Max	Unit	Conditions[2]
Input Low Voltage	VIL	0		0.3*VDDIO	V	
Input High Voltage	VIH	0.7*VDDIO		VDDIO	V	
Hysteresis (Schmitt Trigger) P0.15-P0.8, -RESET	VHS	0.25			V	
Output Low Current	IOL					
P2		6			mA	VOL=0.4V,VDDIO=1.62V
		22			mA	VOL=1.0V,VDDIO=3.00V
P1, P0		4			mA	VOL=0.4V,VDDIO=1.62V
		10			mA	VOL=1.0V,VDDIO=3.00V
D, A, -CS1, -CS0, -RD, -WR, EX, TDO		3			mA	VOL=0.4V,VDDIO=1.62V
		7			mA	VOL=1.0V,VDDIO=3.00V
Output High Current	IOH					
P2		6			mA	VOH=1.2V,VDDIO=1.62V
		11			mA	VOH=2.6V,VDDIO=3.00V
P1, P0		4			mA	VOH=1.2V,VDDIO=1.62V
		7			mA	VOH=2.6V,VDDIO=3.00V
D, A, -CS1, -CS0, -RD, -WR, EX, TDO		3			mA	VOH=1.2V,VDDIO=1.62V
		5			mA	VOH=2.6V,VDDIO=3.00V
	RPU					
Pull-up Resistance P2, P1, P0			11.5, 260, Hi-Z		Kohms	Software selectable
D, A			260, Hi-Z		Kohms	Software selectable
TDI, TMS, TDO, -RESET, -XM, -EMU			130		Kohms	
-CS1, -CS0, -RD, -WR			260, Hi-Z		Kohms	RPU only active during Reset
Pull-down Resistance	RPD					
TEST1, TEST0, -TRST, TCK			11.5		Kohms	
Input Leakage Current	IIL	-10	<1	10	uA	

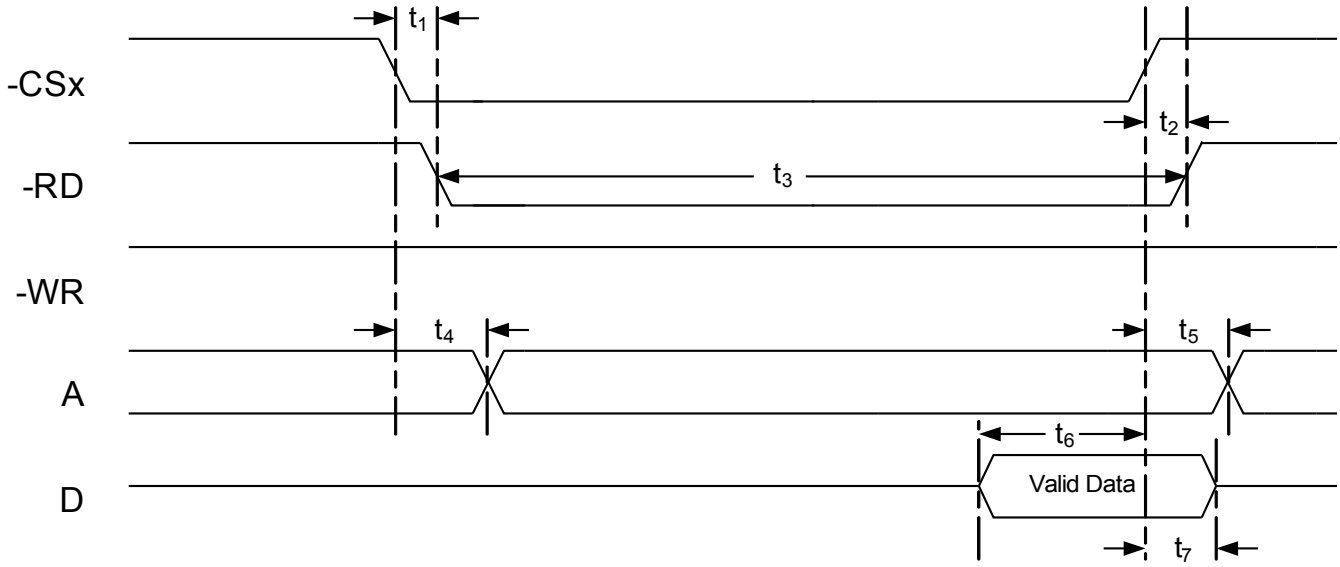
DC Characteristics: USB I/O

Characteristic	Sym	Min	Typ[1]	Max	Unit	Conditions
Differential Receiver Input Sensitivity	VDI	0.2			V	
Differential Receiver Common Mode Voltage	VCM	0.8		2.5	V	
Single Ended Receiver Input Low Voltage	VIL			0.8	V	
Single Ended Receiver Input High Voltage	VIH	2.0			V	
Output Low Voltage	VOL			0.3	V	RL=1.5K ohms to 3.6V
Output High Voltage	VOH	2.8			V	RL=15K ohms to Gnd
High-Z State Data Line Leakage	ILO	-10		+10	uA	
Driver Output Resistance	ZDRV	28		44	ohms	RS=17.4 ohms

AC Characteristics: Digital I/O

Characteristic	Sym	Min	Typ[1]	Max	Unit	Conditions[2]
Output Rise Time	TPLH					
P2				4.5	ns	
P1, P0				4.5	ns	
D, A, -CS1, -CS0, -RD, -WR, EX, TDO				5.0	ns	
Output Fall Time	TPHL					
P2				4.5	ns	
P1, P0				4.5	ns	
D, A, -CS1, -CS0, -RD, -WR, EX, TDO				5.0	ns	

AC Characteristics: External Memory Read Timing (CSRD=RDCS=0)

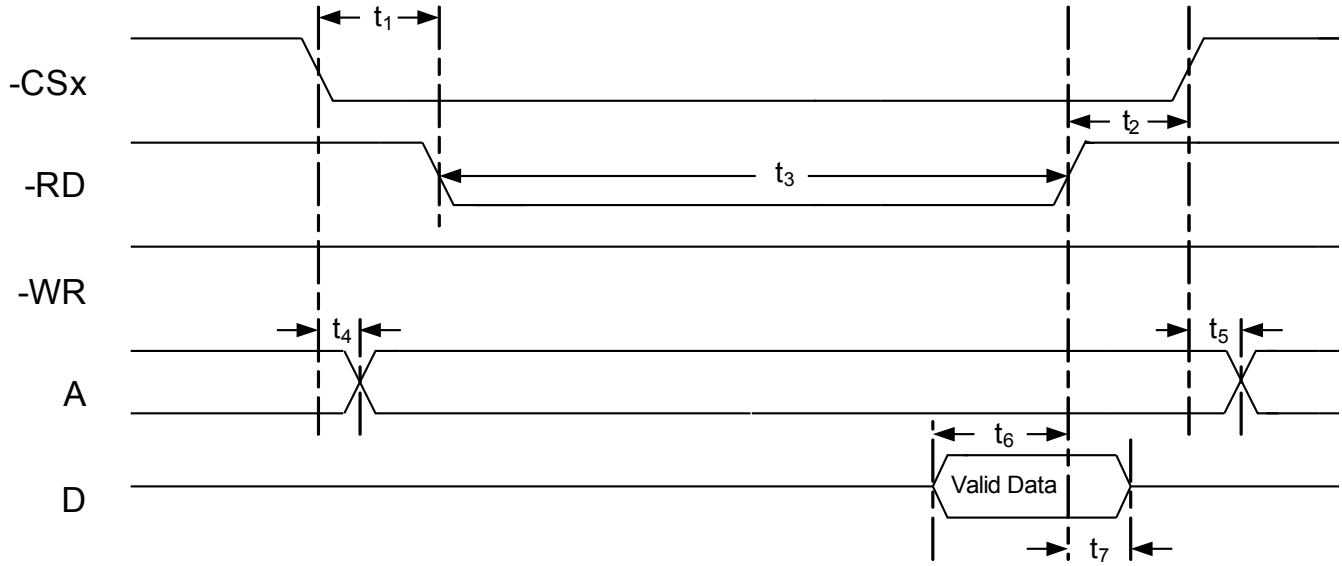


External Memory Read Timing Values (CSRD=RDCS=0)

Parameter	Symbol	Min	Max	Unit
-CSx Low to -RD Low	t_1		0.6	ns
-CSx High to -RD High	t_2		0.4	ns
-RD Pulse Width (-RD Low to -RD High)	t_3	T*(2 + WS)		ns
-CSx Low to A (address) Valid	t_4		3.1	ns
-CSx High to A (address) Changing	t_5	0.7		ns
D (data) Setup to -CSx High	t_6	14.8		ns
D (data) Hold Time (-CSx High to D Invalid)	t_7	0		ns

Notes:
1. CSRD, RDCS, and WS are external memory interface configuration bits in the xmConfig register.
2. In non-Turbo mode, T is equal to the CPUCLK clock cycle (ns). In Turbo mode, T is equal to 2*CPUCLK clock cycle (ns). For example, in non-Turbo mode 36 MHz CPUCLK operation or in Turbo mode 72 MHz CPUCLK operation, T=27.8 ns. These are typical clock rates used by Fluent Chip technologies.

AC Characteristics: External Memory Read Timing (CSRD=RDCS=1)



External Memory Read Timing Values (CSRD=RDCS=1)

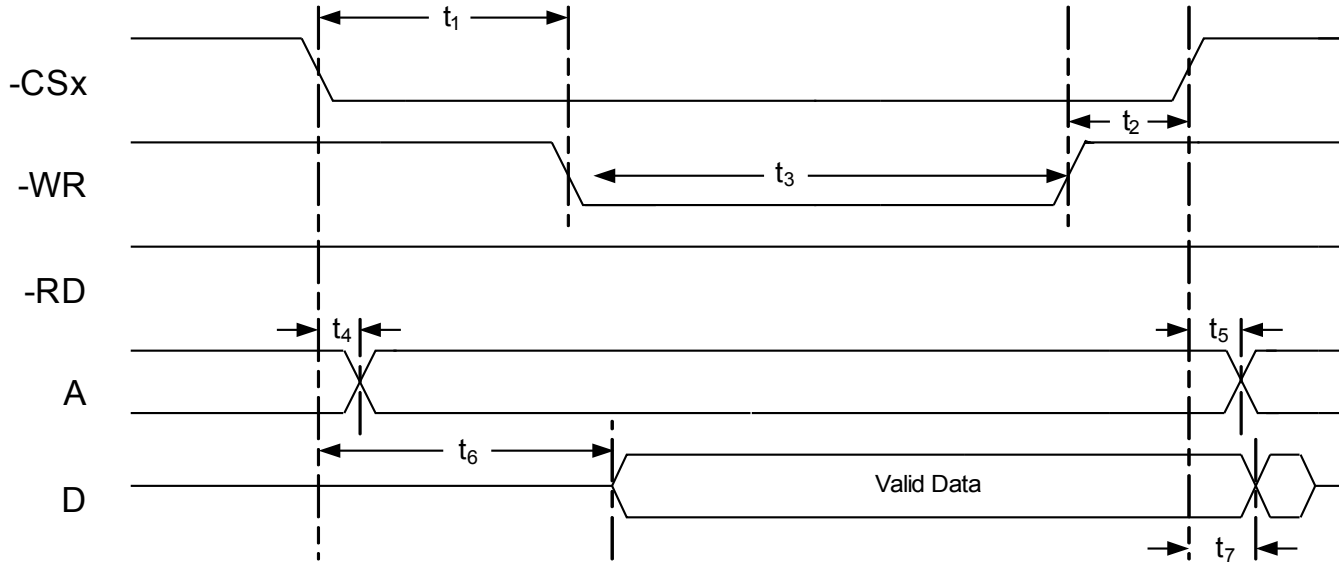
Parameter	Symbol	Min	Max	Unit
-CSx Low to -RD Low	t_1		T	ns
-RD High to -CSx High	t_2		T	ns
-RD Pulse Width (-RD Low to -RD High)	t_3	T*(2 + WS)		ns
-CSx Low to A (address) Valid	t_4		3.1	ns
-CSx High to A (address) Changing	t_5	0.7		ns
D (data) Setup to -RD High	t_6	15.2		ns
D (data) Hold Time (-RD High to D Invalid)	t_7	0		ns

Notes:
3. CSRD, RDCS, and WS are external memory interface configuration bits in the xmConfig register.
4. In non-Turbo mode, T is equal to the CPUCLK clock cycle (ns). In Turbo mode, T is equal to 2*CPUCLK clock cycle (ns). For example, in non-Turbo mode 36 MHz CPUCLK operation or in Turbo mode 72 MHz CPUCLK operation, T=27.8 ns. These are typical clock rates used by Fluent Chip technologies.

External memory access time requirements for 72 MHz (Turbo Mode) or 36 MHz (non-Turbo Mode) operation as a function of the WS bits are listed in the table below. These are typical clock rates used by Fluent Chip technologies.

WS	# of MEMCLKs	Memory Tacc Required (ns)
0	2	37
1	3	65
2	4	93
3	5	121
4	6	148
5	7	176
6	8	204
7	9	232

AC Characteristics: External Memory Write Timing



External Memory Write Timing Values

Parameter	Symbol	Min	Max	Unit
-CSx Low to -WR Low	t_1		2*T	ns
-WR High to -CSx High	t_2		T	ns
-WR Pulse Width (-WR Low to -WR High)	t_3	T*(2 + WS)		ns
-CSx Low to A (address) Valid	t_4		3.1	ns
-CSx High to A (address) Changing	t_5	0.7		ns
-CSx Low to D (data) Valid	t_6		2*T	ns
D (data) Hold Time (-CSx High to D Invalid)	t_7	0		ns

Notes:
5. WS are external memory interface configuration bits in the xmConfig register.
6. In non-Turbo mode, T is equal to the CPUCLK clock cycle (ns). In Turbo mode, T is equal to 2*CPUCLK clock cycle (ns). For example, in non-Turbo mode 36 MHz CPUCLK operation or in Turbo mode 72 MHz CPUCLK operation, T=27.8 ns. These are typical clock rates used by Fluent Chip technologies.

Regulator Characteristics

Characteristic	Min	Typ	Max	Unit	Conditions
Input Voltage	2.0	3.0	3.6	V	
Maximum Current Output			90	mA	Normal Mode
			250	uA	Very Low Power Mode
Output Voltage	1.62	1.8	1.98	V	Normal Mode
	1.37	1.68	1.98	V	Very Low Power Mode
Quiescent Current		35	100	uA	Normal Mode
		1		uA	Very Low Power Mode

POR Block Characteristics

Characteristic	Min	Typ	Max	Unit	Conditions
RESET Hold Timer	10	35	60	mSec	
Low Voltage Threshold		0.89[*1]		DVDDA[*1]	
Brown-Out Reset Threshold		0.84[*1]		DVDDA[*1]	

*1: Thresholds expressed a fraction of DVDDA voltage with the high power regulator enabled.

Oscillator Characteristics

Characteristic	Min	Typ	Max	Unit	Conditions
OSC1 Frequency	2.0	4.0	8.0	MHz	
OSC1 Operating Current		150		uA	OSCVDD = 3.0V
		45		uA	OSCVDD = 1.8V
OSC2 Frequency		32.768		kHz	
OSC2 Operating Current		7		uA	OSCVDD = 3.0V
		2		uA	OSCVDD = 1.8V
OSC3 (RC) Frequency			400	kHz	Uncalibrated after reset
	122	128	134	kHz	After software calibration
OSC3 (RC) Operating Current		7		uA	

PWM Characteristics

All tests done with an 8-ohm load connected between PWM0 and PWM1.

Characteristic	Min	Typ	Max	Unit	Conditions
Voh		2.20		V	PWMVDD = 3.0V
		1.25		V	PWMVDD = 1.8V
Vol		0.55		V	PWMVDD = 3.0V
		0.4		V	PWMVDD = 1.8V
Square-wave Output Power		360		mW	PWMVDD = 3.0V
		90		mW	PWMVDD = 1.8V

DAC Characteristics

Characteristic	Min	Typ	Max	Unit	Conditions
DAC Resolution		16		bits	
Clock Frequency			48	MHz	
Conversion Rate			96	kHz	
Output Range (averaged)	1/4	1/2	3/4	DACVDD	
Idle output noise 20Hz-20KHz relative to full-scale sine wave	-86			dB	DACVDD = 3.0V
THD			0.1	%	
Output Resistance, DACOUTP1/2, DACOUTN1/2		6		kOhm	

ADC Characteristics

Characteristic	Min	Typ	Max	Unit	Conditions
ADC Resolution		16		bits	
Clock Frequency		4		MHz	
Conversion Rate		48	96	kHz	
VCM Output Voltage		0.53		AVDD	
VCM Output Resistance		18		kOhms	
ADC Input Voltage Range	VCM +/- 0.90 Typ.			V	Voltage Range 2[*1]
	VCM +/- 0.675 Typ.			V	Voltage Range 1[*1]
	VCM +/- 0.45 Typ.			V	Voltage Range 0[*1]
Idle noise		-70[*2]		dB	Voltage Range 2[*1]
		-70[*2]		dB	Voltage Range 1[*1]
		-66[*2]		dB	Voltage Range 0[*1]
THD			0.1	%	

*1: Voltage Range 2 = 3.3V typical
Voltage Range 1 = 2.40 to 3.0V typical
Voltage Range 0 = 1.8V typical

*2: Zero input signal, sample rate is 48KHz, decimated to 16KHz prior to measurement. Noise is measured from 20 Hz to 8KHz. Idle noise is relative to a full-scale sinewave.

PGA Characteristics

Characteristic	Min	Typ	Max	Unit	Conditions
MICP1, MICP2 input resistance		20		kOhm	
MICN1, MICN2 resistance to opamp inverting terminal		3.6		kOhm	Line input gain range
		1.1		kOhm	Mic input gain range
AMPOUT1, AMPOUT2 output resistance		1.5		KOhm	
Input signal range			125	mVpp	Centered around VCM
			1.25	Vpp	Centered around VCM
Operating Current		2.5		mA	Per PGA unit
Offset voltage referred to op-amp input			2	mV	

Absolute Maximum Ratings (†)

Operating Temperature	-40C to +85C
Storage Temperature	-65C to +150C
Supply Voltage (VDD, DVDDA, VDDOTP)	-0.3V to +2.2V
Supply Voltage (VDDIO, AVDD, OSCVDD, DACVDD, PWMVDD, USBVDD, VDDIN)	-0.3V to +3.9V
Supply Voltage (VPP)	-0.3V to +7.0V
Voltage on input pins with respect to VSS pins[1]	-0.3V to +3.9V

1. “VSS pins” includes VSS, DVSSA, VSSOTP, VSSIO, AVSS, OSCVSS, DACVSS, PWMVSS, USBVSS

†WARNING: Stresses beyond the “Absolute Maximum Ratings” may cause permanent damage. These are stress ratings only and functional operation of the device at these conditions is not implied. Operation beyond those listed in the “Recommended Operating Conditions” section is not recommended and extended exposure beyond the “Recommended Operating Conditions” may affect device reliability.

The Interactive Speech™ Product Line

Sensory's **Interactive Speech™** product line makes consumer electronics more intelligent by enabling them to talk, hear, move and interact with the external world using naturally sounding spoken commands-all without training and even in noisy environments! Sensory offers both chip and software solutions that offer advanced speech recognition with hands-free functionality, biometric speaker verification, text-to-speech (TTS) synthesis, high quality stereo music and sound effects, robotics and LCD controls, and interactive sensing capabilities. These technologies are designed for integration into cost-sensitive consumer electronic applications such as home appliances, smart toys, music players and personal communication devices. The hardware line includes the NLP-5x Natural Language Processor, the RSC-4x family of mixed signal processors, and the SC-691 music and speech synthesis slave processor. Embedded software options include the FluentSoft™ Recognizer, which offers speech recognition technologies for non-Sensory processors and DSPs. Sensory's BlueGenie™ Voice Interface, the first speech recognition, TTS and synthesis option for *BlueTooth*® enabled devices, offers hands-free control of headsets, music players and other *BlueTooth*® devices.

NLP-5x Natural Language Processor and Development Tools

The NLP-5x features a high-performance 80MHz 16-bit DSP with on-chip ADC, hi-fidelity stereo DAC, microphone preamplifiers, RAM, OTP code and constant memory, and many kinds of peripheral interfaces and control blocks. With Sensory's FluentChip™ 5 firmware, it provides a single chip solution capable of accurate speech recognition; text-to-speech (TTS) synthesis with morphing; compressed speech; high fidelity music; motor and LCD control; and man-machine interfaces (MMI) with interactive sensors. Sensory offers a complete suite of evaluation and development tools that include the ability to create complex grammars with a natural language interface in multiple languages.

RSC-4x Family of Microcontrollers and Developer Tools

The RSC-4x (**Recognition, Synthesis and Control**) product family contains low-cost 8-bit speech-optimized microcontrollers that are fully integrated and include A/D, pre-amplifier, D/A, RAM, and ROM circuitry. With Sensory's FluentChip™ firmware, the RSC family offers speech recognition, speaker verification, speech and music synthesis, voice recording and playback, and an entire suite of interactive robotic and sonic networking technologies. The family is supported by a complete suite of evaluation and development toolkits that include the ability to quickly create speaker independent recognition sets in many languages.

SC6 Slave Processor and Tools

The SC-691 is a standard slave synthesizer that accepts compressed speech data from other microprocessors or microcontrollers and converts it to speech. The chip operates up to 12.32 MIPS, and provides high-quality, low data-rate speech compression and MIDI music synthesis, with unlimited speech duration using external memory. Sensory offers hardware and software tools for analyzing speech files, editing speech data and generating coded speech.

FluentSoft™ Recognizer

The FluentSoft™ Recognizer is the engine powering the FluentSoft™ SDK. It provides a noise-robust, large-vocabulary, speaker-independent solution with continuous digit recognition and word-spotting capabilities. This small-footprint software recognizes thousands of words and runs on non-Sensory processors including Intel XScale, TI OMAP, and ARM9, and supports operating systems such as MS Windows, Linux, and Symbian.

BlueGenie™ Voice Interface

The BlueGenie Voice Interface software suite runs on CSR's BC-5 MM Kalimba DSP, and enables manufacturers of *Bluetooth* products to integrate full voice control and synthetic speech output without the need for visual displays or complex user interfacing. It frees designers to pack functionality onto small form factor *Bluetooth* devices and answers consumer demand for a "Truly Hands-Free" experience.

Important notices:

Safety Policy:

SENSORY®

4701 Patrick Henry Dr., Santa Clara, CA 95054
Tel: **(408) 625-3300** Fax: **(408) 625-3350**