Innovasic®
Semiconductor
Extended Life Semiconductor Solutions

IA70C20
8-Bit Microcontroller
Data Sheet

Published by Innovasic Semiconductor, Inc.
3737 Princeton Drive NE, Suite 130, Albuquerque, NM 87107

TABLE OF CONTENTS

1. Features

- Pin-for-pin compatible with Texas Instruments TMS70C20 CMOS 8-Bit Microcontroller.
- Register-to-register architecture.
- Up to 4K bytes On-Chip ROM.
- 128 bytes Internal RAM.
- 13-Bit Timer.
- Memory-mapped ports for easy adressing.
- Eight Addressing formats.
- Single-instruction BCD add and subtract.
- Two external maskable interrupts.
- Two power-down modes.
  - Wake-up (160□ A at 1 MHz typical)
  - Halt, Xtal/clkin = gnd (1 □ A typical)
- CMOS technology.
- Operating Voltage: 5V $^{+}$/- 10 %.
- Operating Temperature: Industrial Range (-40^{o}C to +85^{o}C).
- Maximum Osc Frequency: 5 MHz.
- Available packages:
  - 44-pin Plastic Leaded Chip Carrier package (PLCC).
  - 40-pin, 600 mil, dual in-line package (DIP).

The IA70CX0 is a form, fit, and function replacement for the original Texas Instruments TMS70CX0 CMOS 8-Bit Microcontroller. The IA70CX0 incorporates a CPU, memory, bit I/ O, timer, interrupts and external bus interface logic on a single chip. Typical applications for the IA70CX0 microcontroller include industrial, consumer, computer, telecom and automotive applications. InnovASIC's version of the microcontroller includes all the features listed above and is "plug and play" with the original Texas Instruments device.

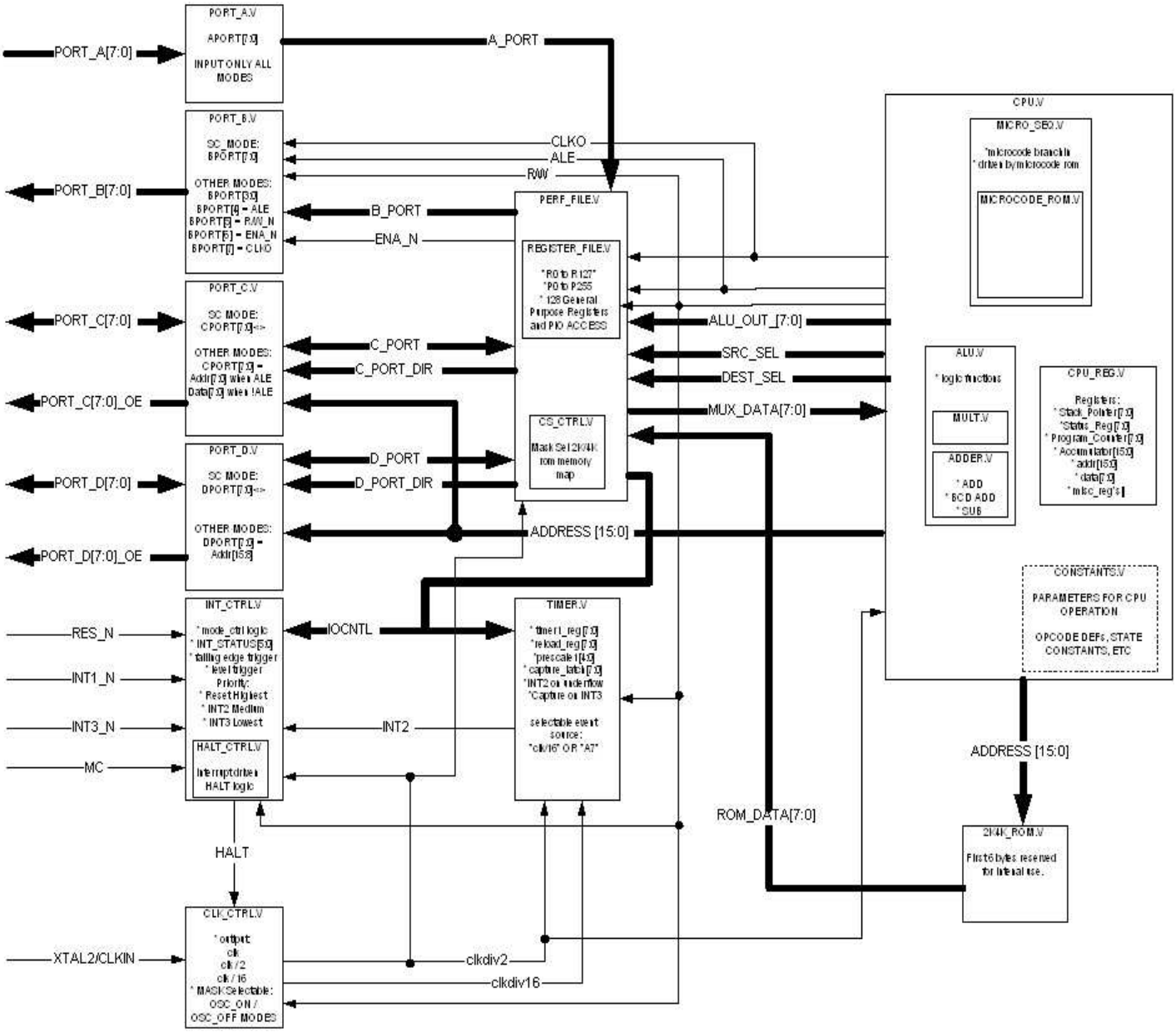
InnovASIC produces replacement ICs using its MILES^{TM}, or Managed IC Lifetime Extension System, cloning technology. This technology produces replacement ICs far more complex than "emulation" while ensuring they are compatible with the original IC. MILES^{TM} captures the design of a clone so it can be produced even as silicon technology advances. MILES^{TM} also verifies the clone against the original IC so that even the "undocumented features" are duplicated.

2. Description

The IA70C20 microcontroller replaces obsolete TI® TMS70C20 devices, allowing customers to retain existing board designs, software compilers/assemblers, and emulation tools, thereby avoiding expensive redesign efforts.

A block diagram of the 70C20 microcontroller is depicted in Figure 1.

Figure 1. IA70C20 Block Diagram



The 70C20 microcontroller consists of the following functional blocks:

- CPU
- Port A
- Port B
- Port C
- Port D
- Interrupt Controller
- Clock Controller
- Perf File
- Timer
- ROM

A brief description of each block follows:

2.1 CPU

The CPU block contains the microcode sequencer, the ALU, and the CPU registers. The microcode sequencer controls the process of reading and executing the microcode that enables the IA70C20 to execute assembly code. The ALU performs all of the logical and arithmetical operations for the device as required by the microcode. The CPU registers block contains basic information about the function of the IA70C20. The two registers are the 16 bit program counter (PC) and the 8 bit stack pointer (SP).

2.2 Port A

Port A is an 8 bit input only port. Pin A7 has a second function as the clock for the on-chip Timer/Event counter.

2.3 Port B

Port B is an 8-bit output port. Pins B3-B0 are general purpose bits while pins B7-B4 are dual function pins. When in single-chip mode these pins are general purpose bits, otherwise they are bus control bits.

2.4 Port C

In single-chip mode, Port C is an 8-bit bi-directional port in which each pin may be individually set to be either an input or an output under the control of software. In all other modes, Port C becomes a multiplexed address/data port for the off-chip memory bus providing the least significant byte of a 16-bit address.

2.5 Port D

In either single-chip or peripheral expansion mode, Port D is an 8-bit bi-directional port in which each pin may be individually set to be either an input or an output under the control of software. In either full expansion or microprocessor mode, Port D contains the most significant byte of the 16-bit address.

2.6 Interrupt Controller

There are four interrupt levels INT0-INT3, with INT0 having the highest priority. This block receives the external interrupt and alerts the CPU, enabling servicing of the interrupt. The interrupts are synced to the positive edge of x2_div_2, the divided clock.

2.7 Clock Controller

Generates two enable pulse signals, one at ½ x2, and one at 1/16 x2, to clock internal registers at varying speeds.

2.8 Perf File

Contains register file registers, data holding registers, and peripheral registers for port operations.

2.9 Timer

A programmable timer/event counter. It is an 8 bit modulo-n counter with a programmable pre-scaled clock source. INT2 is an internal interrupt used by the timers.

2.10 ROM

Customer specific 2K X 8 instruction ROM.

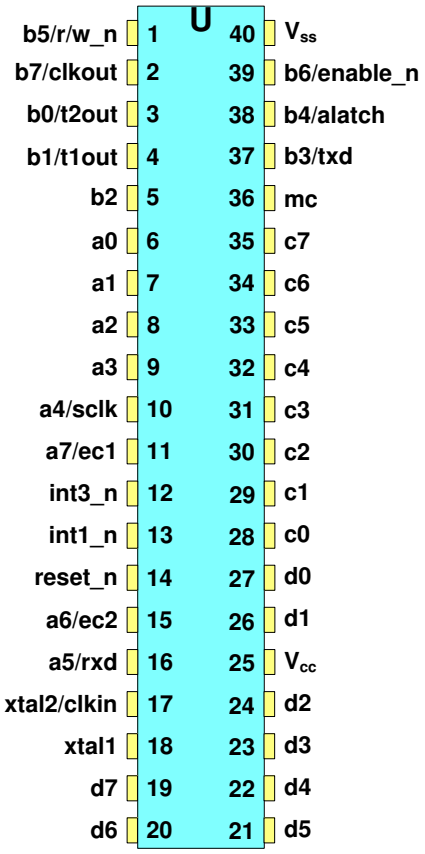
Hexadecimal Instruction Table/Opcode Map

High		0000 0	0001 1	0010 2	0011 3	0100 4	0101 5	0110 6	0111 7	1000 8	1001 9	1010 A	1011 B	1100 C	1101 D	1110 E	1111 F
LOW 0000	0	NOP								MOVP Pn,A			TSTA/ CLRC	MOV A,B	MOV A,Rn	JMP	TRAP 15
0001	1	IDLE									MOVP Pn,B			TSTB	MOV B,Rn	JN/ JLT	TRAP 14
0010	2		MOV Rn,A	MOV %n,A	MOV Rn,B	MOV Rn,Rn	MOV %,n,B	MOV B,A	MOV %n,Pn	MOVP A,Pn	MOVP Pn,B	MOVP %n,Pn	DEC A	DEC B	DEC Rn	JZ/ JEQ	TRAP 13
0011	3		AND Rn,A	AND %n,A	AND Rn,B	AND Rn,Rn	AND %,n,B	AND B,A	AND %n,Pn	ANDP A,Pn	ANDP B,Pn	ANDP %n,Pn	INC A	INC B	INC Rn	JC/ JHS	TRAP 12
0100	4		OR Rn,A	OR %n,A	OR Rn,B	OR Rn,Rn	OR %,n,B	OR B,A	OR %n,R	ORP A,Pn	ORP B,Pn	ORP %n,Pn	INV A	INV B	INV Rn	JP/ JGT	TRAP 11
0101	5	EINT	XOR Rn,A	XOR %n,A	XOR Rn,B	XOR Rn,Rn	XOR %,n,B	XOR B,A	XOR %n,R	XORP A,Pn	XORP B,Pn	XORP %n,Pn	CLR A	CLR B	CLR Rn	JPZ/ JGE	TRAP 10
0110	6	Dint	BTJO Rn,A	BTJO %n,A	BTJO Rn,B	BTJO Rn,Rn	BTJO %,n,B	BTJO B,A	BTJO %n,R	BTJOP A,Pn	BTJOP B,Pn	BTJOP %n,Pn	XCHB A	XCHB B	XCHB Rn	JNZ/ JNE	TRAP 9
0111	7	SETC	BTJZ Rn,A	BTJZ %n,A	BTJZ Rn,B	BTJZ Rn,Rn	BTJZ %,n,B	BTJZ B,A	BTJZ %n,R	BTJZP A,Pn	BTJZP B,Pn	BTJZP %n,Pn	SWAP A	SWAP B	SWAP Rn	JNC/ JL	TRAP 8
1000	8	POP ST	ADD Rn,A	ADD %n,A	ADD Rn,B	ADD Rn,Rn	ADD %,n,B	ADD B,A	ADD %n,R	MOVD %n,Rn	MOVD Rn,Rn	MOVD %n,(B), Rn	PUSH A	PUSH B	PUSH Rn	TRAP 23	TRAP 7
1001	9	STSP	ADC Rn,A	ADC %n,A	ADC Rn,B	ADC Rn,Rn	ADC %,n,B	ADC B,A	ADC %n,R				POP A	POP B	POP Rn	TRAP 22	TRAP 6
1010	A	RETS	SUB Rn,A	SUB %n,A	SUB Rn,B	SUB Rn,Rn	SUB %,n,B	SUB B,A	SUB %n,R	LDA @n	LDA *Rn	LDA @n(B)	DJNZ A	DINZ B	DINZ Rn	TRAP 21	TRAP 5
1011	B	RETI	SBB Rn,A	SBB %n,A	SBB Rn,B	SBB Rn,Rn	SBB %,n,B	SBB B,A	SBB %n,R	STA @n	STA *Rn	STA @n(B)	DECD A	DECD B	DECD Rn	TRAP 20	TRAP 4
1100	C		MPY Rn,A	MPY %n,A	MPY Rn,B	MPY Rn,Rn	MPY %,n,B	MPY B,A	MPY %n,R	BR @n	BR *Rn	BR @n(B)	RR A	RR B	RR Rn	TRAP 19	TRAP 3
1101	D	LDSP	CMP Rn,A	CMP %n,A	CMP Rn,B	CMP Rn,Rn	CMP %,n,B	CMP B,A	CMP %n,R	CMPA @n	CMPA *Rn	CMPA @n(B)	RRC A	RRC B	RRC Rn	TRAP 18	TRAP 2
1110	E	PUSH ST	DAC Rn,A	DAC %n,A	DAC Rn,B	DAC Rn,Rn	DAC %,n,B	DAC B,A	DAVC %n,R	CALL @n	CALL *Rn	CALL @n(B)	RL A	RL B	RL Rn	TRAP 17	TRAP 1
1111	F		DSB Rn,A	DSB %n,A	DSB Rn,B	DSB Rn,Rn	DSB %,n,B	DSB B,A	DSB %n,R				RLC A	RLC B	RLC Rn	TRAP 16	TRAP 0

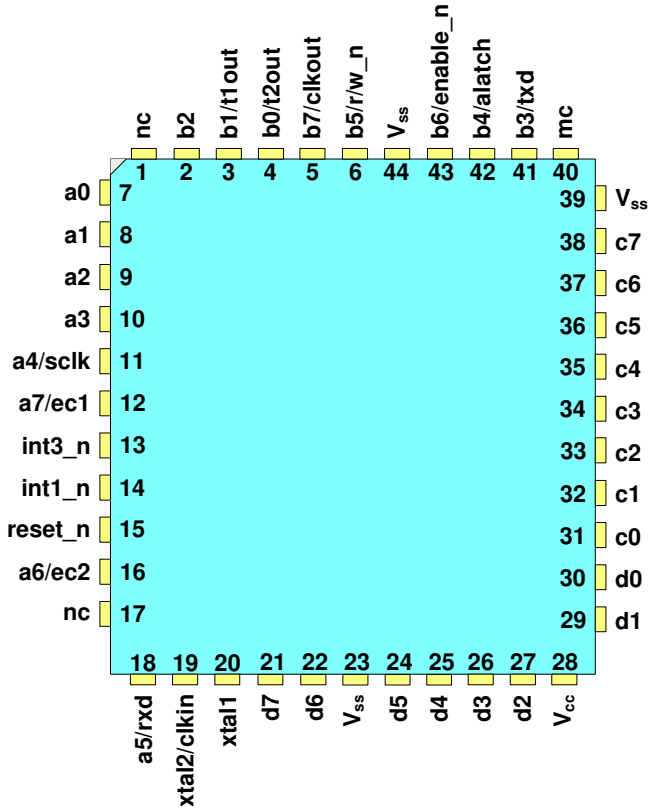
A - Register A Rn - Register File Register %n - Immediate Addressing *Rn - Indirect Addressing
B - Register A Pn - Peripheral File Register @n - Direct Addressing

Signal	Pin		I/O	Description
	PLCC	DIP		
A0	7	6	I	Port A. All pins may be used as high-impedance input-only lines. Pin A7/EC1 may also be used as the timer/event counter input.
A1	8	7	I	
A2	9	8	I	
A3	10	9	I	
A4	11	10	I	
A5	18	16	I	
A6	16	15	I	
A7/EC1	12	11	I	
B0	3	3	O	Port B. B0-B7 are general-purpose output-only pins. B4-B7 become memory-expansion control signals in peripheral-expansion, full-expansion, and microprocessor modes.
B1	4	4	O	
B2	5	5	O	
B3	41	37	O	
B4/ALATCH	42	38	O	Data output/memory interface address latch strobe.
B5/R/W	1	1	O	Data output/memory read/write signal.
B6/ENABLE	43	39	O	Data output/memory interface enable strobe.
B7/CLKOUT	2	2	O	Data output/internal clockout.
C0	31	28	I/O	Port C. C0-C7 can be individually selected in software as general-purpose input or output pins in single-chip mode. C0-C7 become the LSB address/data bus in peripheral-expansion, full-expansion, and microprocessor modes.
C1	32	29	I/O	
C2	33	30	I/O	
C3	34	31	I/O	
C4	35	32	I/O	
C5	36	33	I/O	
C6	37	34	I/O	
C7	38	35	I/O	
D0	30	27	I/O	Port D. D0-D7 can be individually selected in software as general-purpose input or output pins in single-chip or peripheral-expansion modes. D0-D7 become the MSB address/data bus in full-expansion and microprocessor modes.
D1	29	26	I/O	
D2	27	24	I/O	
D3	26	23	I/O	
D4	25	22	I/O	
D5	24	21	I/O	
D6	22	20	I/O	
D7	21	19	I/O	
INT1	14	13	I	Highest priority maskable interrupt
INT3	13	12	I	Lowest priority maskable interrupt
RESET	15	14	I	Device reset
MC	40	36	I	Mode control pin, VCC for microprocessor mode
XTAL2/CLKIN	19	17	I	Crystal input for control of internal oscillator
XTAL1	20	18	O	Crystal ouput for control of internal oscillator
VCC	28	25		Supply voltage (positive)
VSS	44 39 23	40		Ground reference

* Also apply to SE70CP160A prototyping device.



A. Plastic 40-Pin DIP



B. 44-Pin PLCC

3. Addressing Modes

IA70C20 assembly language supports eight addressing modes. These eight modes are listed in Table 1. A description of each mode follows the table.

Table 1. IA70C20 Addressing Modes

Addressing Mode	Example		
Single register	LABEL	DEC	B
		INC	R45
		CLR	R23
Dual register	LABEL	MOV	B , A
		ADD	A , R17
		CMP	R32 , R73
Peripheral file	LABEL	XORP	A , R17
		MOVP	P42 , B
Immediate	LABEL	AND	%>C5 , R55
		ANDP	%VALUE , P32
		BTJO	%>D6 , R80 , LABEL

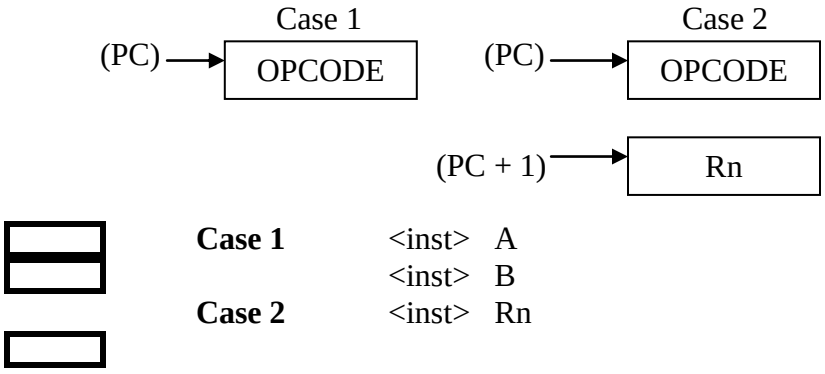
Program counter relative	LABEL 1	JMP DJNZ BTJO BTJOP	LABEL A , LABEL %>16 , R12 , LABEL B , P7 , LABEL
Direct memory	LABEL	LDA CMPA	@>F3D4 @LABEL
Register file indirect	LABEL	STA	*R43
Indexed	LABEL 2	BR	@LABEL (B)

3.1 Single-Register Addressing Mode

In single-register addressing mode, a single register denoted by Rn (n is the register file number in the range 0-127) containing an eight-bit operand is used. A and B can denote R0 and R1, respectively.

Figure 1 illustrates the object code generated by a single operand instruction for the following cases:

Figure 1. Single-Register Addressing Mode Object Code



3.2 Dual-Register Addressing Mode

In dual register addressing mode, instructions use a source and a destination register that each contain 8-bit operands. The source register is always listed prior to the destination register in the assembly language. Figure 2 illustrates the byte requirements for all dual addressing mode instructions.

Figure 2. Dual-Register Addressing Mode Byte Requirements

Destination

Source	A	B	Rd
A		1	2
B	1		2
iop	2	2	3
Rs	2	2	3

Bytes Needed for Move Instructions

Destination

Source	A	B	Rd
A		2	3
B	1		3
iop	2	2	3
Rs	2	2	3

Bytes Needed for All Other Instructions

3.3 Peripheral-File Addressing Mode

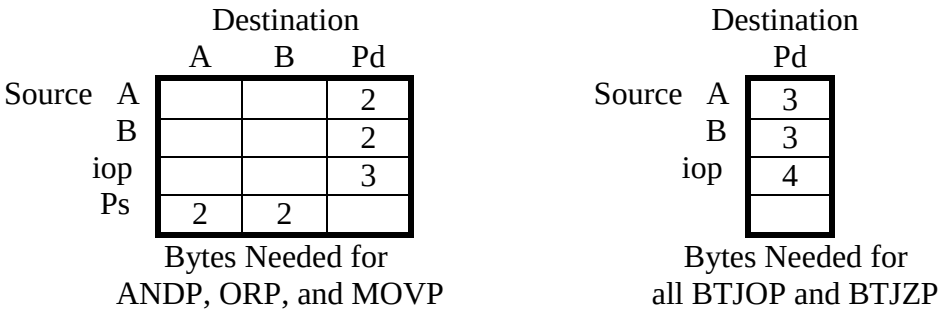
In peripheral-file addressing mode, instructions perform I/O tasks. Each PF register is an 8-bit port that can be referred to as Pn.

Four instructions use peripheral-file addressing mode:

- MOVP,
- ANDP,
- ORP, and
- XORP.

These instructions may use register A or B as the source register and Pn as the destination register. MOVP may also be executed using Pn as the source register and A or B as the destination register. (BTJOP and BTJZP are also peripheral-file instructions but they have a different format.) Figure 3 illustrates the byte requirements of the instructions using the peripheral-file addressing mode.

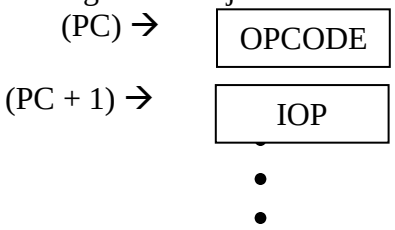
Figure 3. Peripheral-File Addressing Mode Byte Requirements



3.4 Immediate Addressing Mode

In immediate addressing mode instructions use an immediate eight-bit operand. Either a constant value or a label preceded by a percent sign (%) can be used as the immediate value. The MOVD instruction uses 16-bit immediate operands in two special formats. Figure 4 illustrates the simplest case of an instruction using this mode.

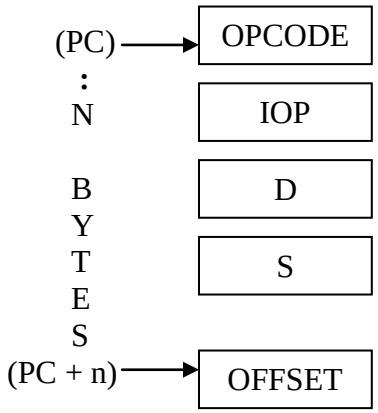
Figure 4. Immediate Addressing Mode Object Code



3.5 Program Counter Relative Addressing Mode

All jump instructions use program counter relative addressing mode. A target address (ta) must be included in any assembly language jump instruction. The offset is calculated as follows: offset = ta – pcn, where **pcn** is the location of the next instruction and –128 ≤ ta ≤ 127. Figure 5 illustrates object code generated by a jump instruction.

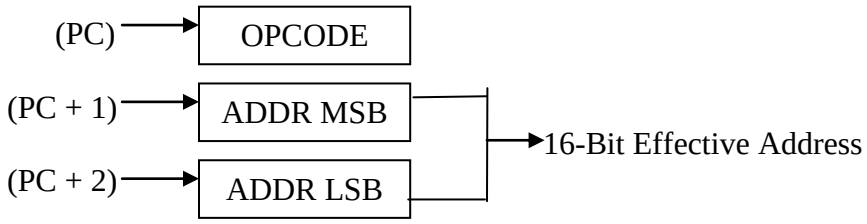
Figure 5. Program Counter Relative Addressing Mode Object Code



3.6 Direct Memory Addressing Mode

The operand for a direct addressing mode instruction is located in memory. The location is indicated by a 16-bit address. The 16-bit address is preceded by an @ sign and can be written as a constant value or as a label. Figure 6 shows how the object code produced by an instruction using the direct memory addressing mode generates a 16-bit effective address.

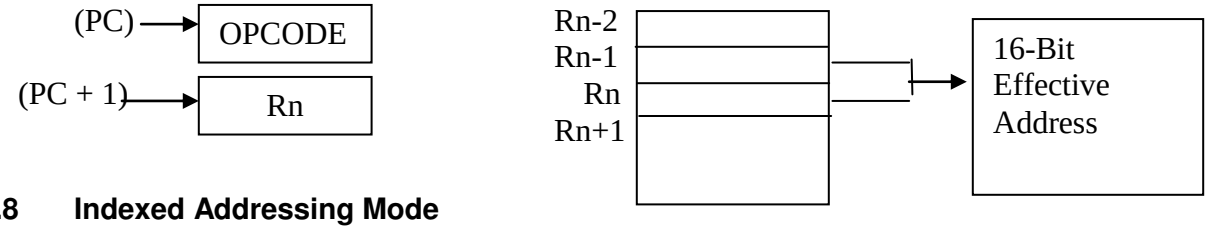
Figure 6. Direct Memory Addressing Mode Object Code



3.7 Register-File Indirect Addressing Mode

A register pair containing a 16-bit effective address is used in register file indirect addressing mode. The indirect register file address is written as a register number (Rn) preceded by an asterisk (*), that is, *Rn. The LSB of the address is contained in Rn, and the MSB of the address is contained in the previous register (Rn-1). Figure 7 shows how the object code produced by an instruction using register file indirect addressing mode generates a 16-bit effective address.

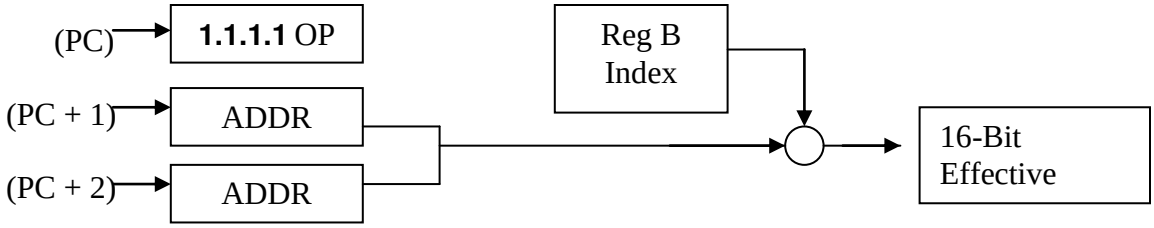
Figure 7. Register-File Indirect Addressing Mode Object Code



3.8 Indexed Addressing Mode

In indexed addressing mode, a 16-bit address formed by adding the contents of the B register to a 16-bit direct memory address is used to access the operand. The assembly language statement for the indexed addressing mode contains the direct memory address written as a 16-bit constant value or a label, preceded by an @ sign and followed by a B in parentheses: @LABEL(B). The addition automatically transfers any carries into the MSB. Figure 8 illustrates how the object code produced by an instruction using the indexed addressing mode generates a 16-bit effective address.

Figure 8. Indexed Addressing Mode Object Code



4. Instruction Overview

Following is a listing of assembly language instructions for the IA70C20. Labels, mnemonics, operands, and comments must be separated by at least one space in the assembly code:

TMS7000 Family Instruction Overview

Mnemonic	Opcode	Bytes	Cycles Tc(C)	Status C N Z I	Operation Description
ADC B,A	69	1	5	R R R x	(s) + (Rd) + (C) → (Rd)
Rs,A	19	2	8		Add the source, destination, and
Rs,B	39	2	8		carry bit together. Store at the
Rs,Rd	49	3	10		destination.
%iop,A	29	2	7		
%iop,B	59	2	7		

	%iop,Rd	79	3	9		
ADD	B,A	68	1	5	R R R x	(s) + (Rd) → (Rd) Add the source and destination operands at the destination address.
	Rs,A	18	2	8		
	Rs,B	38	2	8		
	Rs,Rd	48	3	10		
	%iop,A	28	2	7		
	%iop,B	58	2	7		
	%iop,Rd	78	3	9		
AND	B,A	63	1	5	0 R R x	(s) AND (Rd) → (Rd) AND the source and destination operands together and store at the destination address.
	Rs,A	13	2	8		
	Rs,B	33	2	8		
	Rs,Rd	43	3	10		
	%iop,A	23	2	7		
	%iop,B	53	2	7		
	%iop,Rd	73	3	9		
ANDP	A,Pd	83	2	10	0 R R x	(s) AND (Pd) → (Pd) AND the source and destination operands together and store at the destination address.
	B,Pd	93	2	9		
	%iop,Pd	A3	3	11		

Note: Add two to cycle count if branch is taken

Legend:

0	Status bit set always to 0.
1	Status bit set always to 1
R	Status bit set to a 1 or a 0 depending on results of operation.
x	Status bit not affected.
b	Bit ( ) affected.
Ofst	Offset

TMS7000 Family Instruction Overview (Continued)

Mnemonic	Opcode	Bytes	Cycles Tc(C)	Status C N Z I	Operation Description
BR @Label @Label(B) *Rn	8C AC 9C	3 3 2	10 12 9	x x x x	(XADDR) → (PC) The PC will be replaced with the contents of the destination operand.
(1) BTJO B,A,Ofst Rn,A,Ofst Rn,B,Ofst Rn,Rd,Ofst %iop,A,Ofst %iop,B,Ofst %iop,Rn,Ofst	 66 16 36 46 26 56 76	 2 3 3 4 3 3 4	 7 (9) 10 (12) 10 (12) 12 (14) 9 (11) 9 (11) 11 (13)	0 R R x	If (s [Bit x]) AND (Rn [Bit x]) ≠ 0, then (PC) + offset → (PC) If the AND of the source and destination operands ≠ 0, the PC will be modified to include the offset.
(1) BJOP A,Pn,Ofst B,Pn,Ofst %>iop,Pn,Ofst	 86 96 A6	 3 3 4	 11 (13) 10 (12) 12 (14)	0 R R x	If (s [Bit x]) AND (Pn [Bit x]) ≠ 0, then (PC) + offset → (PC) If the AND of the source and destination operands ≠ 0, the PC will be modified to include the offset.
(1) BTJZ B,A,Ofst Rn,A,Ofst Rn,B,Ofst Rn,Rf,Ofst %>iop,A,Ofst %>iop,B,Ofst %>iop,Rn,Ofst	 67 17 37 47 27 57 77	 2 3 3 4 3 3 4	 7 (9) 10 (12) 10 (12) 12 (14) 9 (11) 9 (11) 11 (13)	0 R R x	If (s [Bit x]) AND NOT(Rn [Bitx]) ≠ 0, then (PC) + offset → (PC) If the AND of the source and NOT(destination) operands ≠ 0, the PC will be modified to include the offset.
(1) BTJZP A,Pn,Ofst B,Pb,Ofst %>iop,Pb,Ofst	 87 97 A7	 3 3 4	 11 (13) 10 (12) 12 (14)	0 R R x	If (s [Bit x]) AND NOT(Pn [Bitx]) ≠ 0, then (PC) + offset → (PC) If the AND of the source and NOT(destination) operands ≠ 0, the PC will be modified to include the offset.
CALL @Label @Label(B) *Rn	8E AE 9E	3 3 2	14 16 13	x x x x	(SP) + 1 → (SP) (PC MSB) → ((SP)) (SP) + 1 → (SP) (PC LSB) → ((SP)) (XADDR) → (PC)
CLR A B Rd	B5 C5 D5	1 1 2	5 5 7	0 0 1 x	0 → (Rd) Clear the destination operand.
CLRC	B0	1	6	0 R R x	0 → (C) Clears the carry bit.

Note: Add two to cycle count if branch is taken

Legend:

0 Status bit set always to 0.
1 Status bit set always to 1
R Status bit set to a 1 or a 0 depending on results of operation.
x Status bit not affected.
b Bit ( ) affected.
Ofst Offset

TMS7000 Family Instruction Overview (Continued)

Mnemonic	Opcode	Bytes	Cycles Tc(C)	Status C N Z I	Operation Description
CMP B,A	6D	1	5	R R R x	(Rn) – (s) computed but not stored Set flags on the result of the source operand subtracted from the destination operand.
Rn,A	1D	2	8		
Rn,B	3D	2	8		
Rn,Rn	4D	3	10		
%iop,A	2D	2	7		
%iop,B	5D	2	7		
%iop,Rn	7D	3	9		
CMPA @Label	8D	3	12	R R R x	(A) – (XADDR) computed but not stored Set flags on result of the source operand subtracted from A.
@Label(B)	AD	3	14		
*Rn	9D	2	11		
DAC B,A	6E	1	7	R R R x	(s) + (Rd) + (C) → (Rd) (BCD) The source, destination, and the carry bit are added, and the BCD sum is stored at the destination address. Contents on the s + Rd operands initially need to be the BCD.
Rs,A	1E	2	10		
Rs,B	3E	2	10		
Rs,Rd	4E	3	12		
%>iop,A	2E	2	9		
%>iop,B	5E	2	9		
%>iop,Rd	7E	3	11		
DEC A	B2	1	5	R R R x	(Rd) – 1 → (Rd) Decrement destination operand by 1.
B	C2	1	5		
Rd	D2	2	7		
DECD A	BB	1	9	R R R x	(Rd) – 1 → (Rp) Decrement register pair by 1.C=0 on 0 – FFFF transition
B	CB	1	9		
Rp	DB	2	11		
DINT	06	1	5	0 0 0 0	0 → (global interrupt enable bit). Clear the I bit.
(1)				x x x x	
DJNZ A,Ofst	BA	1	7 (9)		(Rd) – 1 → (Rd); If (Rd) ≠ 0, (PC) + offset → (PC)
B,Ofst	CA	2	7 (9)		
Rd,Ofst	DA	2	9 (11)		
DSB B,A	6F	1	7	R R R x	(Rd) – (s) – 1 + (C) → (Rd) (BCD) The source of the operand is subtracted from the destination; this sum is then reduced by 1 and the carry bit is then added to it. The result is stored as a BCD number. Contents on the s + Rd operands initially need to be BCD.
Rs,A	1F	2	10		
Rs,B	3F	2	10		
Rs,Rd	4F	3	12		
%>iop,A	2F	2	9		
%>iop,B	5F	2	9		
%>iop,Rd	7F	3	11		
EINT	05	1	5	1 1 1 1	2 → (global interrupt enable bit). 3 Set the I bit.
IDLE	01	1	6	x x x x	(PC) → (PC) until interrupt (PC) + 1 → (PC) after return from interrupt Stops µC execution until an interrupt.

Note: Add two to cycle count if branch is taken

Legend:

0 Status bit set always to 0.
1 Status bit set always to 1
R Status bit set to a 1 or a 0 depending on results of operation.
x Status bit not affected.
b Bit ( ) affected.
Ofst Offset

TMS7000 Family Instruction Overview (Continued)

Mnemonic	Opcode	Bytes	Cycles Tc(C)	Status C N Z I	Operation Description
INC A	B3	1	5	R R R x	(Rd) + 1 → (Rd)
B	C3	1	5		Increase the destination operand by 1
Rd	D3	2	7		
INV A	B4	1	5	0 R R x	NOT(Rd) → (Rd)
B	C4	1	5		1's complement the destination operand.
Rd	D4	2	7		
JMP Ofst	D0	2	7	x x x x	(PC) + offset → (PC)
					The PC is modified by an offset to create a new PC value.
(1)				x x x x	
JC Ofst	E3	2	5 (7)		If conditions are met, then (PC) + offset
JEQ Ofst	E2	2	5 (7)		→ (PC)
JHS Ofst	E3	2	5 (7)		If the needed conditions are met, the PC
JL Ofst	E7	2	5 (7)		is modified by the offset to form a new
JN Ofst	E1	2	5 (7)		PC value.
JNC Ofst	E7	2	5 (7)		
JNE Ofst	E6	2	5 (7)		
JNZ Ofst	E6	2	5 (7)		
JP Ofst	E4	2	5 (7)		
JPZ Ofst	E5	2	5 (7)		
JZ Ofst	E2	2	5 (7)		
LDA @Label	8A	3	11	0 R R x	(XADDR) → (A)
@Label(B)	AA	3	13		Move the source operand to A.
*Rn	9A	2	10		
LDSP	0D	1	5	x x x x	(B) → (SP)
					Load SP with register B's contents.
MOV A,B	C0	1	6	0 R R x	(s) → (Rd)
A,Rd	D0	2	8		Replace the destination operand with the
B,A	62	1	5		source operand.
B,Rd	D1	2	7		
Rs,A	12	2	8		
Rs,B	32	2	8		
Rs,Rd	42	3	10		
%>iop,A	22	2	7		
%>iop,B	52	2	7		
%>iop,Rd	72	3	9		
MOVD %>iop,Rp	88	4	15	0 R R x	(s) → (Rd)
%>iop(B),Rp	A8	4	17		Copy the source register pair to the
Rp,Rp	98	3	14		destination register pair.
MOVP A,Pd	82	2	10	0 R R x	(s) → (Pd) or (Ps) → (d)
B,Pd	92	2	9		Copy the source operand into the
%>iop,Pd	A2	3	11		destination operand.
Ps,A	80	2	9		
Ps,B	91	2	8		

Note: Add two to cycle count if branch is taken

Legend:

0	Status bit set always to 0.
1	Status bit set always to 1
R	Status bit set to a 1 or a 0 depending on results of operation.
x	Status bit not affected.
b	Bit ( ) affected.
Ofst	Offset

TMS7000 Family Instruction Overview (Continued)

Mnemonic	Opcode	Bytes	Cycles Tc(C)	Status C N Z I	Operation Description
MPY B,A Rs,A Rs,B Rn,Rn %>iop,A %>iop,B %>iop,Rn	6C 1C 3C 4C 2C 5C 7C	1 2 2 3 2 2 3	44 47 47 49 46 46 48	0 R R x	(s) x (Rn) → (A,B) Multiply the source and destination operands, store the result in registers A (MSB) and B (LSB).
NOP	00	1	4	x x x x	(PC) + 1 → (Rd) Add 1 to the PC
OR B,A Rs,A Rs,B Rd,Rd %>iop,A %>iop,B %>iop,Rd	64 14 34 44 24 54 74	1 2 2 3 2 2 3	5 8 8 10 7 7 9	0 R R x	(s) OR (Rd) → (Rd) Logically OR the source and destination operands, and store the results at the destination address.
ORP A,Pd B,Pd %>iop,Pd	84 94 A4	2 2 3	10 9 11	0 R R x	(s) OR (Pd) → (Pd) Logically OR the source and destination operands, and store the results at the destination address.
POP A B Rd	B9 C9 D9	1 1 2	6 6 8	0 R R x	((SP)) → (Rd) (SP) – 1 → (SP) Copy the last byte on the stack into the destination address.
POP ST	08	1	6	Loaded from stack	((SP)) → (ST) (SP) – 1 → (SP) Replace the status register with the last byte of the stack.
PUSH A B Rs	B8 C8 D8	1 1 2	6 6 8	x x x x	(SP) + 1 → (SP) (Rs) → (SP) Copy the operand onto the stack.
PUSH ST	0E	1	6	x x x x	(SP) + 1 → (SP) (Status register) → ((SP)) Copy the status register onto the stack.
RETI	0B	1	9	Loaded from stack	((SP)) → (PC) LSByte (SP) –1 → (SP) ((SP)) → (PC) MSByte (SP) –1 → (SP) ((SP)) → status register (SP) –1 → (SP)
RETS	0A	1	7	x x x x	((SP)) → (PC LSB) (SP) –1 → (SP) ((SP)) → (PC MSB) (SP) –1 → (SP)

Note: Add two to cycle count if branch is taken

Legend:

0 Status bit set always to 0.
1 Status bit set always to 1
R Status bit set to a 1 or a 0 depending on results of operation.
x Status bit not affected.
b Bit ( ) affected.
Ofst Offset

TMS7000 Family Instruction Overview (Continued)

Mnemonic	Opcode	Bytes	Cycles Tc(C)	Status C N Z I	Operation Description
RL A	BE	1	5	b7 R R x	Bit(n) → Bit(n + 1)
B	CE	1	5		Bit(7) → Bit(0) and Carry
Rd	DE	2	7		
RLC A	BF	1	5	b7 R R x	Bit(n) → Bit(n + 1)
B	CF	1	5		Carry → Bit(0)
Rd	DF	2	7		But(7) → Carry
RR A	BC	1	5	b0 R R x	Bit(n +1) → Bit(n)
B	CC	1	5		Bit(0) → Bit(7) and Carry
Rd	DC	2	7		
RRC A	BD	1	5	b0 R R x	Bit(n +1) → Bit(n)
B	CD	1	5		Carry → Bit(7)
Rd	DD	2	7		Bit(0) → Carry
SBB B,A	6B	1	5	R R R x	(Rd) – (s) –1 + (C) → (Rd)
Rs,A	1B	2	8		Destination minus source minus 1 plus
Rs,B	3B	2	8		carry; stored at the destination address.
Rs,Rd	4B	3	10		
%>iop,A	2B	2	7		
%>iop,B	5B	2	7		
%>iop,Rd	7B	3	9		
SETC	07	1	5	1 0 1 x	1→ (C) Set the carry bit.
STA @Label	8B	3	11	0 R R R x	(A) → (XADDR)
@Label(B)	AB	3	13		Store A at the destination.
*Rd	9B	2	10		
STSP	09	1	6	x x x x	(SP) → (B) Copy the SP into register B.
SUB B,A	6A	1	5	R R R x	(Rd) – (s) → (Rd)
Rs,A	1A	2	8		Store the destination operand minus the
Rs,B	3A	2	8		source operand into the destination.
Rs,Rd	4A	3	10		
%>iop,A	2A	2	7		
%>iop,B	5A	2	7		
%>iop,Rd	7A	3	9		
SWAP A	B7	1	8	R R R x	Rd(Hn,Ln) → Rd(Ln,Hn)
B	C7	1	8		Swap the operand's hi and lo nibbles.
Rn	D7	2	10		
TRAP 0-23	E8-FF	1	14	x x x x	(SP) + 1 → (SP) (PC MSB) → ((SP)) (SP) +1 → (SP) (PC LSB) → ((SP)) (Entry vector) → (PC)

Note: Add two to cycle count if branch is taken

Legend:

0 Status bit set always to 0.
1 Status bit set always to 1
R Status bit set to a 1 or a 0 depending on results of operation.
x Status bit not affected.
b Bit ( ) affected.
Ofst Offset

TMS7000 Family Instruction Overview (Continued)

Mnemonic	Opcode	Bytes	Cycles Tc(C)	Status C N Z I	Operation Description
TSTA	B0	1	6	0 R R x	0 → (C) Set carry bit; set sign and zero flags on the value of register A.
TSTB	C1	1	6	0 R R x	0 → (C) Set carry bit; set sign and zero flags on the value register B.
XCHB A Rn	B6 D6	1 2	6 8	0 R R x	(B) ←→ (Rn) Swap the contents of register B with (d).
XOR B,A Rs,A Rs,B Rs,Rd %>iop,A %>iop,B %>iop,Rd	65 15 35 45 25 55 75	1 2 2 3 2 2 3	5 8 8 10 7 7 9	0 R R x	(s) XOR (Rd) → (Rd) Logically exclusive OR the source and destination operands, store at the destination address.
XORP A,Pd B,Pd %>iop,Pd	85 95 A5	2 2 3	10 9 11	0 R R x	(s) XOR (Pd) → (Pd) Logically exclusive OR the source and destination operands, store at the destination.

Note: Add two to cycle count if branch is taken

Legend:

0	Status bit set always to 0.
1	Status bit set always to 1
R	Status bit set to a 1 or a 0 depending on results of operation.
x	Status bit not affected.
b	Bit ( ) affected.
Ofst	Offset

5. DC Characteristics

Ta = -40°C to Ta = 85°C, Supply voltage = 4.5V to 5.5Vdc unless otherwise specified.

Parameter	Symbol	Conditions	Limits		Unit
			Min	Max	
Input Voltage High	Vih	All Except INT1, INT3, RESET_N	0.7*Vcc	Vcc+0.3	V
Input Voltage Low	Vil	All Except INT1, INT3, RESET_N	Vcc-0.3	0.3*Vcc	V
Input Voltage High	Vih	INT1, INT3, RESET_N	3.22	Vcc+0.3	V
Input Voltage Low	Vil	INT1, INT3, RESET_N	Vcc-0.3	1.84	V
Input Capacitance	Ci	Not Tested - Guaranteed by process	-	4	pf
Output Voltage High	Voh	Ioh = 6mA	Vcc-0.5	-	V
		Ioh = 24mA			
Output Voltage Low	Vol	Iol = 10mA	Vcc-2.0	-	V
Supply Current	Icc	Vcc = 5V, fosc(min) 1.35Mhz fosc(max) = 5Mhz	9.4	35.0	ma
Wake Up Current	Iccwu	Not Measured	-	-	-
Halt Current Device CLK Active	Icch	Not Measured	-	-	-
Halt Current Device CLK Stopped	Icchs	Not Measured	-	-	-
Input Leakage Current	Icchs	Vcc = 5V	-	< 10	ua
Tri-State Leakage Current	Ii	4.5 < Vcc < 5.5	-	10	ua
Low Voltage Operation	Itsl	Normal Operating Conditions	4.5	5.5	V

6. AC Characteristics

Ta = -40 °C to Ta = 85 °C, Supply voltage = 4.5V to 5.5 Vdc unless otherwise specified.
Reference figures 3.4.2.2,3.4.2.3,3.4.2.4,3.4.2.5, for interface timing relationships.

Parameter	Symbol	Conditions	Limits		Unit
			Min	Max	
RC Network Frequency	fosc	R = 15k, C = 47pF,	1.4	2.1	MHz
RC Network Osc Input	Vihrc		0.7*Vcc	-	V
	Vilrc		-	0.3*Vcc	V
RC Clock to CLKOUTA Delay	td	Rtest = 1k, Ctest = 1pf		28	ns
Clockout External Cycle Time	tc(c)	tc(c) = 2/fosc	952	1481	
Clock Internal State Cycle Time	tc(s)	tc(s) = 2/fosc	952	1481	
Crystal Cycle Time	tc(p)	tc(p) = 1/fosc	741	476	
Clockin Pulse Width High	tf	45% to 55% of 1/fosc	333	214	
Clockin Pulse Width Low	tw	55% to 45% of 1/fosc	333	214	
Clockout Pulse Width High	tw(ch)	Caution! This device is to be used in the single chip mode only. Other operating modes are used for functional testing purposes. These characteristics would be used for expansion (Memory) Modes and are not applicable to the single chip mode.			
Clockout Pulse Width Low	tw(cl)				
Clockout Rise to ALATCH Fall	td(ch-jl)				
ALATCH Pulse Width High	tw(jh)				
Address Valid High before ALATCH Fall	td(ah-jl)				
Address Valid Low before ALATCH Fall	td(al-jl)				
Address HOLD Low before ALATCH Fall	th(jl-al)				
R/W Valid before ALATCH Fall	td(rw-jl)				

Signal Name	req. setup time	measured setup time	req. hold time	measured hold time	req. clock to out	measured clock to out
pa_0	80	1.38	70	1.38	na	na
pa_1	80	1.35	70	1.35	na	na
pa_2	80	1.33	70	1.33	na	na
pa_3	80	1.29	70	1.29	na	na
pa_4	80	1.22	70	1.22	na	na
pa_5	80	1.15	70	1.15	na	na
pa_6	80	1.14	70	1.14	na	na
pa_7	80	1.19	70	1.19	na	na
pb_0	na	na	na	na	100	24.27
pb_1	na	na	na	na	100	24.33
pb_2	na	na	na	na	100	24.31
pb_3	na	na	na	na	100	24.28
pb_4	na	na	na	na	100	25.83
pb_5	na	na	na	na	100	25.95
pb_6	na	na	na	na	100	25.71
pb_7	na	na	na	na	100	26.68
pc_0	80	39.06	70	39.06	100	56.05
pc_1	80	36.01	70	38.15	100	56.91
pc_2	80	38.3	70	38.74	100	56.54
pc_3	80	36.42	70	38.53	100	56.35
pc_4	80	39.91	70	39.91	100	57.27
pc_5	80	39.81	70	39.94	100	57.77
pc_6	80	39.14	70	39.99	100	58.28
pc_7	80	39.26	70	39.26	100	56.53
pd_0	80	1.18	70	1.18	100	26.27
pd_1	80	1.15	70	1.15	100	26.13
pd_2	80	1.13	70	1.13	100	26.03
pd_3	80	1.13	70	1.13	100	26.1
pd_4	80	1.14	70	1.14	100	26.09
pd_5	80	1.16	70	1.16	100	26.09
pd_6	80	1.15	70	1.15	100	26.44
pd_7	80	1.13	70	1.13	100	26.49
int1_n	na	3.05	na	2.93	na	na
int3_n	na	3.09	na	3.09	na	na
mc	na	24.98	na	45.68	na	na

7. 70cX0 Errata

- When in MC mode, bus contents do not match cycle for cycle outside of qualified data times. IE. When ALE or ENABLE_N are not active, we do not necessarily match.
- When in MC mode, our part does NOT assert RW_N when performing writes to internal registers. OEM part does assert RW_N during internal writes, but not consistently.
- Stack addressing 'underflow' or 'under roll' behavior. Behavior is different between our part vs. oem when an uneven number of 'pop' operations are done for a given number of 'push' operations. If you 'pop' below register file address 0x00 the OEM will stay at 0x00 for the first illegal pop, then go somewhere below 0x00. Given 'n' illegal pop operations it will take 'n-1' push operations to bring stack pointer back to a valid number (0x00). The InnovASIC design will pop down to 0x00 then stop. Any push operations after reaching 0x00 will result in incrementing of stack address.
- Register File addressing – when performing an instruction which manipulates two register file locations such as decrement double on address 0x00 of register file. OEM operates on first byte at 0x00, then decrements to 0xFF which is NOT a valid register file location. Our part decrements to the top of real physical memory 0x7F.

8. Revision History

The table below presents the sequence of revisions to document IA211030117.

Date	Revision	Description	Page(s)
August 19, 2008	05	Corrected control number and reformatted some elements to meet publication standards.	NA