Freescale Semiconductor, Inc. MC68HC05CxRG/AD REV 1

HC05

MC68HC05C4,C8,C9 MC68HC705C8 MC68HC805C4 MC68HCL05C4,C8 MC68HSC05C4,C8

PROGRAMMING REFERENCE GUIDE



The MC68HC05 Family of HCMOS devices covered in this reference guide are as follows:

MC68HC05C4 MC68HC05C8 MC68HC05C9 MC68HC705C8 MC68HC805C4 MC68HCL05C4 MC68HCL05C8 MC68HSC05C4 Freescale Semiconductor, Inc.



MEMORY MAPS

REGISTER/CONTROL BIT ASSIGNMENTS

INSTRUCTIONS ADDRESSING MODES EXECUTION TIMES

MECHANICAL DATA

HEX/DEC CONVERSION ASCII CHART



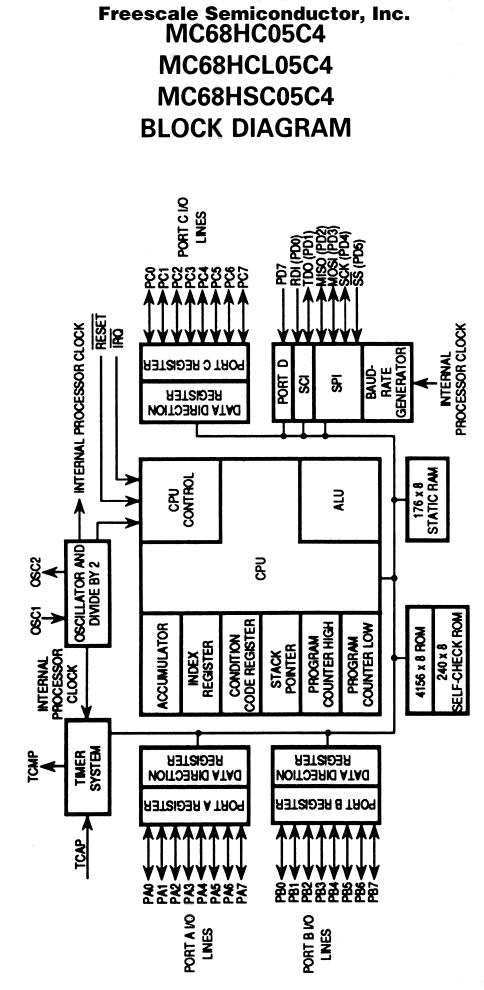
MEMORY MAPS

REGISTER/CONTROL BIT ASSIGNMENTS

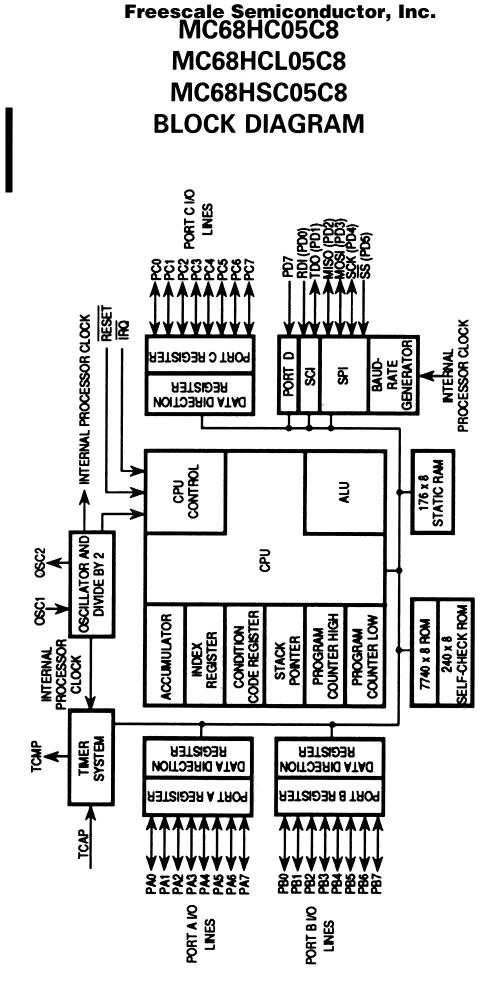
INSTRUCTIONS ADDRESSING MODES EXECUTION TIMES

MECHANICAL DATA

HEX/DEC CONVERSION ASCII CHART

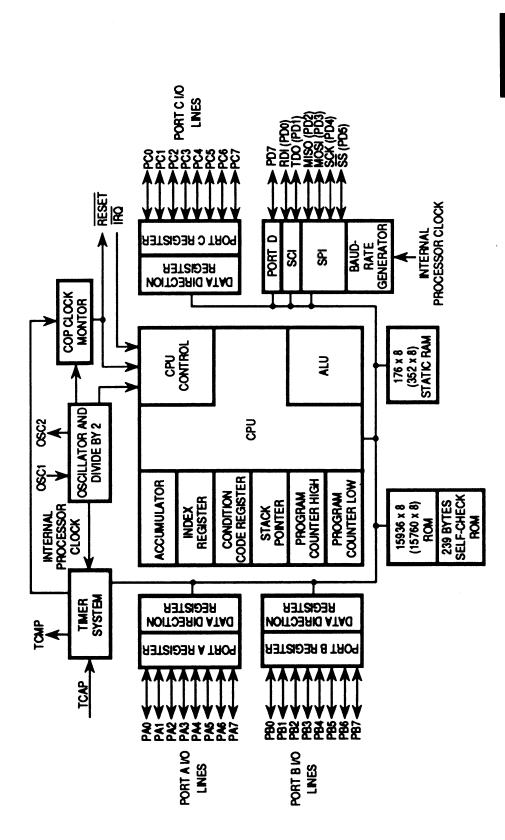


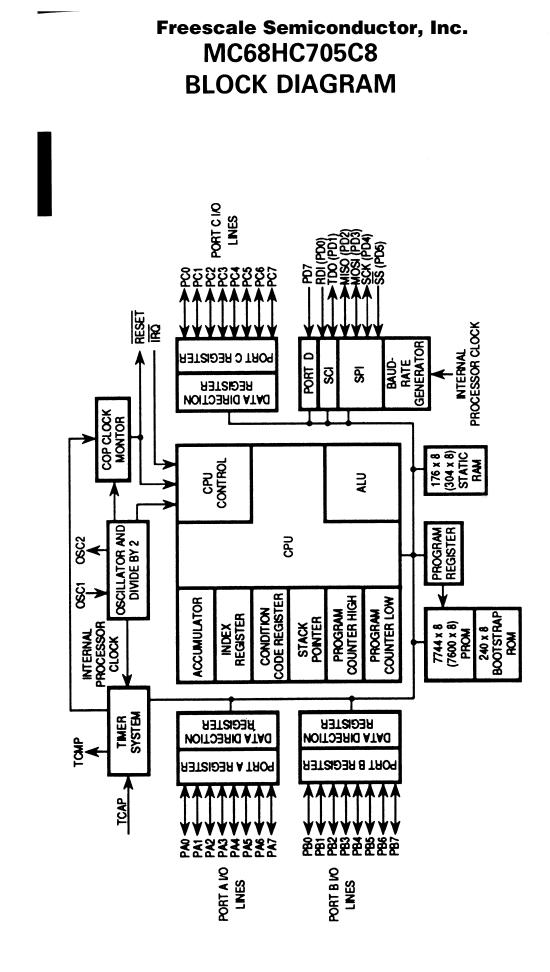


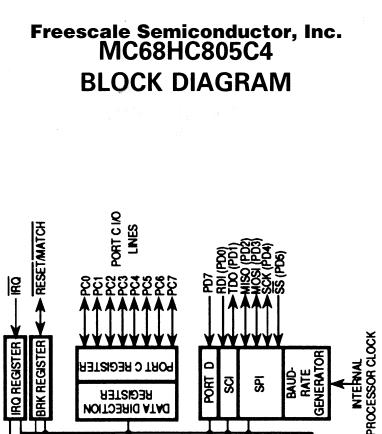


For More Information On This Product, Go to: www.freescale.com

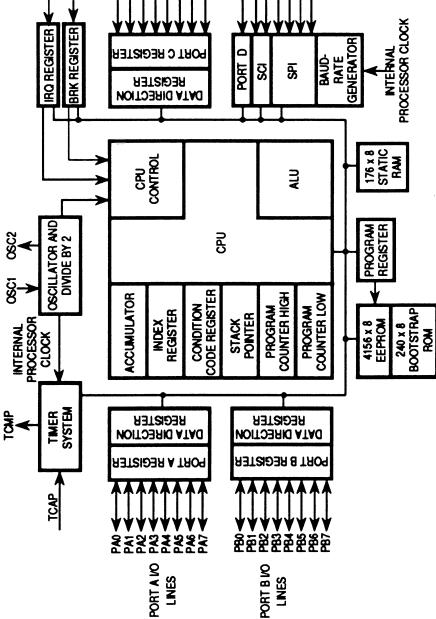
Freescale Semiconductor, Inc. MC68HC05C9 BLOCK DIAGRAM











Freescale Semiconductor, Inc. MC68HC05C4 MC68HCL05C4 MC68HSC05C4 MC68HSC05C4 MEMORY MAP

| \$0000 F | | | 0000 | | ſ | ٦ |
|------------------|----------------|----------|--------------|-------|--|--------------|
| | 1/0 | n | | | PORT A DATA REGISTER | \$00 |
| | | BYTES | | | PORT B DATA REGISTER | \$01 |
| \$001F | | | 0031 | | PORT C DATA REGISTER | \$02 |
| \$0020 | US | | 0032 | | PORT D FIXED INPUT REGISTER | \$ 03 |
| \$004F | RO 48 BY | | 0079 | | PORT A DATA DIRECTION REGISTER | \$04 |
| \$0050 | | | 0080 | | PORT B DATA DIRECTION REGISTER | \$05 |
| | RA | | | 1 | PORT C DATA DIRECTION REGISTER | \$06 |
| \$00BF | 176 B | YTES | 0191 | İ. | UNUSED | \$07 |
| \$00C0 | ` | STACK | 0192 | | UNUSED | \$08 |
| \$00FF | | 64 BYTES | 0255 0256 | | UNUSED | \$09 |
| \$0100 | US | | 0250 | Ì | SERIAL PERIPHERAL CONTROL REGISTER | SOA |
| | RO 2096 B | | | Ì | SERIAL PERIPHERAL STATUS REGISTER | \$0B |
| \$10FF \$1100 | | | 2303 2304 | 1 | SERIAL PERIPHERAL DATA I/O REGISTER | soc |
| 31100 | UNU | | 2304 | | SERIAL COMMUNICATIONS BAUD RATE REGISTER | SOD |
| \$1EFF | 3584 B | IYTES | 7935 | | SERIAL COMMUNICATIONS CONTROL REGISTER 1 | \$0E |
| \$1F00 | | | 7936 | 11 | SERIAL COMMUNICATIONS CONTROL REGISTER 2 | \$0F |
| | SELF-C | HECK | | | SERIAL COMMUNICATIONS STATUS REGISTER | \$10 |
| \$1FDF \$1FE0 | | | | | SERIAL COMMUNICATIONS DATA REGISTER | \$11 |
| | SELF-C VECT | | 8175 | 256 | TIMER CONTROL REGISTER | \$12 |
| \$1FEF \$1FF0 | UNU | | 8176 | BYTES | TIMER STATUS REGISTER | \$ 13 |
| \$1FF3 | 4 BY | | 8179 | | INPUT CAPTURE HIGH REGISTER | \$14 |
| \$1FF4 | USI | | 8180 | | INPUT CAPTURE LOW REGISTER | \$ 15 |
| | VECT 12 BY | | | | OUTPUT COMPARE HIGH REGISTER | \$16 |
| \$1FFF L | | 120 | 8191 | ' | OUTPUT COMPARE LOW REGISTER | \$17 |
| | | | | 1 | COUNTER HIGH REGISTER | \$18 |
| | | | | 1 | COUNTER LOW REGISTER | \$19 |
| | | | | ļ | ALTERNATE COUNTER HIGH REGISTER | S1A |
| | | | | | ALTERNATE COUNTER LOW REGISTER | \$1B |
| | | | | | UNUSED | \$1C |
| | | | | l | UNUSED | \$1D |
| | | | | ļ | UNUSED | \$1E |
| | | | | | UNUSED | SIF |

Freescale Semiconductor, Inc.

Freescale Semiconductor, Inc. MC68HC05C8 **MC68HCL05C8** MC68HSC05C8 **MEMORY MAP**

| | 10000 - | · · | | |
|------------------|--------------|-------|--|------|
| 1/0 | | | PORT A DATA REGISTER | \$00 |
| 32 BYTES | | | PORT B DATA REGISTER | \$01 |
| | 0031 | | PORT C DATA REGISTER | \$02 |
| USER ROM | 0032 | | PORT D FIXED INPUT REGISTER | \$03 |
| 48 BYTES | 0079 | | PORT A DATA DIRECTION REGISTER | \$04 |
| | 0080 | | PORT B DATA DIRECTION REGISTER | \$05 |
| RAM 176 BYTES | | | PORT C DATA DIRECTION REGISTER | \$06 |
| | 0191 | | UNUSED | \$07 |
| | 0192 | | UNUSED | \$08 |
| 64 BYTES | 0255 0256 | | UNUSED | \$09 |
| USER | | | SERIAL PERIPHERAL CONTROL REGISTER | \$0A |
| ROM 680 BYTES | | | SERIAL PERIPHERAL STATUS REGISTER | \$0B |
| <u></u> | 7935 | | SERIAL PERIPHERAL DATA I/O REGISTER | \$0C |
| ELF-CHECK | 1000 | | SERIAL COMMUNICATIONS BAUD RATE REGISTER | \$0D |
| ELF-UNEUN | | | SERIAL COMMUNICATIONS CONTROL REGISTER 1 | \$0E |
| ELF-CHECK | - | | SERIAL COMMUNICATIONS CONTROL REGISTER 2 | SOF |
| VECTORS | 8175 | 256 | SERIAL COMMUNICATIONS STATUS REGISTER | \$10 |
| UNUSED | 8176 | BYTES | SERIAL COMMUNICATIONS DATA REGISTER | \$11 |
| 4 BYTES | 8179 8180 | | TIMER CONTROL REGISTER | \$12 |
| USER VECTORS | 0100 | | TIMER STATUS REGISTER | \$13 |
| 12 BYTES | 8191 | | INPUT CAPTURE HIGH REGISTER | \$14 |
| | | i | INPUT CAPTURE LOW REGISTER | \$15 |
| | | Ì | OUTPUT COMPARE HIGH REGISTER | \$16 |
| | | ļ | OUTPUT COMPARE LOW REGISTER | \$17 |
| | | Ì | COUNTER HIGH REGISTER | \$18 |
| | | | COUNTER LOW REGISTER | \$19 |
| | | 1 | ALTERNATE COUNTER HIGH REGISTER | \$1A |
| | | | ALTERNATE COUNTER LOW REGISTER | \$1B |
| | | | UNUSED | \$1C |
| | | Į | UNUSED | \$1D |
| | | | UNUSED | \$1E |
| | | | UNUSED | \$1F |
| | | | | _ |

| | 1/ | 0 | | |
|------------------|---------|---|-------|--------------|
| | 32 B | YTES | | |
| \$001F | | | 0031 | |
| \$0020 | US | ER | 0032 | |
| | RC | M | İ | |
| \$004F | 48 B | YTES | 0079 | |
| \$0050 | | | 0080 | |
| | 0/ | M | | 1 |
| | | BYTES | | i i |
| \$00BF | 170 2 | ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,, | 0191 | |
| \$00C0 | | STACK | 0192 | 1 |
| \$00FF | | 64 BYTES | 0255 | |
| \$0100 | | | 0256 | i i |
| | US | SER | | |
| | RC | M | | i i |
| e1000 | 7680 | BYTES | 7935 | |
| \$1EFF \$1F00 | <u></u> | | 7935 | |
| 31500 | | | 1300 | |
| | SELF- | CHECK | | |
| | | | | |
| \$1FDF | | СНЕСК | { | i |
| \$1FE0 | | TORS | | |
| \$1FEF | | luns | 8175 | 256 BYTES |
| \$1FF0 | 1 | JSED | 8176 | |
| \$1FF3 | 4 B | YTES | 8179 | |
| \$1FF4 | US | SER | 8180 | I I |
| | | TORS | | |
| \$1FFF | 12 B | INTES | 8191 | I I |
| 31666 | L | | TOIAI | |
| | | | | 1 |

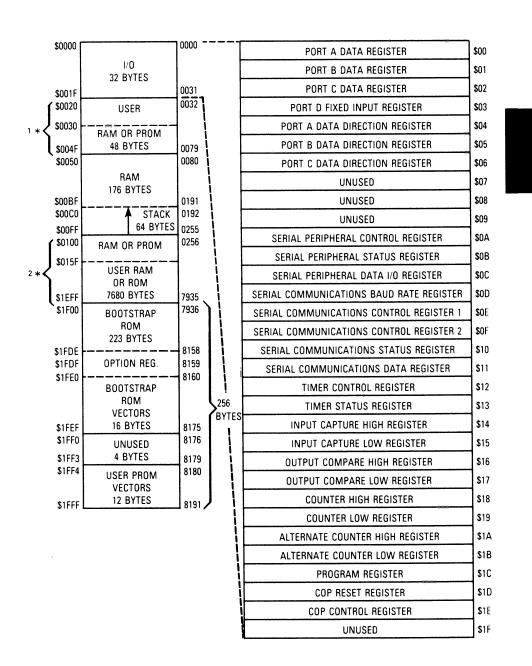
\$0000

Freescale Semiconductor, Inc. MC68HC05C9 MEMORY MAP

| | PORT A DATA REGISTER |] \$ 00 |
|---|--|----------------|
| | PORT B DATA REGISTER | \$01 |
| | PORT C DATA REGISTER | \$ 02 |
| | PORT D DATA REGISTER |] \$ 03 |
| | PORT A DATA DIRECTION REGISTER | 504 |
| | PORT B DATA DIRECTION REGISTER | \$05 |
| | PORT C DATA DIRECTION REGISTER | \$06 |
| | PORT D DATA DIRECTION REGISTER | \$07 |
| | UNUSED | \$08 |
| | UNUSED | \$ 09 |
| | SERIAL PERIPHERAL CONTROL REGISTER | \$04 |
| | SERIAL PERIPHERAL STATUS REGISTER | \$ 08 |
| | SERIAL PERIPHERAL DATA I/O REGISTER | \$00 |
| | SERIAL COMMUNICATIONS BAUD RATE REGISTER | \$00 |
| ļ | SERIAL COMMUNICATIONS CONTROL REGISTER 1 | \$01 |
| Ì | SERIAL COMMUNICATIONS CONTROL REGISTER 2 | \$ 06 |
| i | SERIAL COMMUNICATIONS STATUS REGISTER | \$10 |
| Ì | SERIAL COMMUNICATIONS DATA REGISTER | \$11 |
| Ì | TIMER CONTROL REGISTER | \$12 |
| Ì | TIMER STATUS REGISTER | \$13 |
| | INPUT CAPTURE HIGH REGISTER | \$14 |
| | INPUT CAPTURE LOW REGISTER | \$15 |
| | OUTPUT COMPARE HIGH REGISTER | \$16 |
| | OUTPUT COMPARE LOW REGISTER | \$17 |
| | COUNTER HIGH REGISTER | \$18 |
| | COUNTER LOW REGISTER | \$19 |
| | ALTERNATE COUNTER HIGH REGISTER | \$14 |
| ļ | ALTERNATE COUNTER LOW REGISTER | \$16 |
| ļ | UNUSED | \$10 |
| ļ | COP RESET REGISTER | \$10 |
| ļ | COP CONTROL REGISTER | \$1E |
| | UNUSED | S1F |

| \$0000 | | 00000 |
|--------|---|-------|
| | 1/0 32 BYTES | 00031 |
| \$001F | <u></u> | |
| \$0020 | PAGE ZERO ROM OR RAM 48 BYTES | 00032 |
| \$004F | 40 DTTES | 00079 |
| \$0050 | | 00080 |
| | RAM 176 BYTES | |
| | | 00191 |
| | STACK | 00192 |
| \$00FF | 64 BYTES | 00255 |
| \$0100 | ROM OR RAM 128 BYTES | 00256 |
| \$017F | 120 DTIES | 00383 |
| \$0180 | | 00384 |
| | MAIN MEMORY ROM 15744 ⁻ BYTES | |
| \$3EFF | | 16127 |
| \$3F00 | SELF-CHECK ROM | 16349 |
| \$3FDE | 223 BYTES | 16571 |
| \$3FDF | OPTION REGISTER | 16572 |
| \$3FE0 | | 16573 |
| | SELF-CHECK | |
| | VECTORS | |
| \$3FEF | 16 BYTES | 16588 |
| \$3FF0 | USER | 16589 |
| | VECTORS | |
| \$3FFF | 16 BYTES | |
| a3666 | | 16604 |

Freescale Semiconductor, Inc. MC68HC705C8 MEMORY MAP



(Option Register 1FDF RAM1 = 0 and RAM0 = 0) (POR or Master Reset)

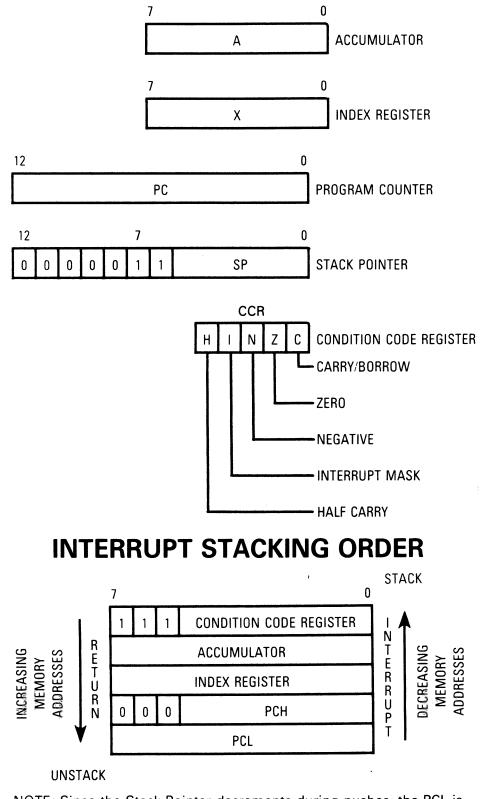
- *The nature of this memory block (RAM or PROM) is controlled by bits RAM0 and RAM1 of the Option Register (\$1FDF).
- 1. RAM0 0 48 Bytes User PROM
- RAM0 1 32 Bytes RAM with 16 Bytes Unused
- 2. RAM1 0 7680 Bytes User PROM
 - RAM1 1 7584 Bytes User PROM and 96 Bytes of RAM

Freescale Semiconductor, Inc. MC68HC805C4 MEMORY MAP

| \$0000 | | 0 | PORT A DATA REGISTER |
|------------------|--------------------------------|-------------------|--|
| | 1/0 32 BYTES | | PORT B DATA REGISTER |
| \$001F | 02 01120 | 0031 | PORT C DATA REGISTER |
| \$0020 | USER | 0032 | PORT D FIXED INPUT REGISTER |
| \$004F | EEPROM 48 BYTES | 0079 | PORT A DATA DIRECTION REGISTER |
| \$004 | 40 01120 | 0080 | PORT B DATA DIRECTION REGISTER |
| | RAM | | PORT C DATA DIRECTION REGISTER |
| \$00BF | 176 BYTES | 0191 | UNUSED |
| \$00C0 | STACK | 0192 | UNUSED |
| \$00FF | 64 BYTE | 0255 | UNUSED |
| 30100 | USER | 0250 | SERIAL PERIPHERAL CONTROL REGISTER |
| | EEPROM 4096 BYTES | | SERIAL PERIPHERAL STATUS REGISTER |
| \$10FF \$1100 | | 4351 4352 | SERIAL PERIPHERAL DATA I/O REGISTER |
| | UNUSED | 4352 | SERIAL COMMUNICATIONS BAUD RATE REGISTER |
| \$1EFF | 3584 BYTES | 7935 | SERIAL COMMUNICATIONS CONTROL REGISTER 1 |
| \$1F00 | BOOTSTRAP | 7936 | SERIAL COMMUNICATIONS CONTROL REGISTER 2 |
| \$1FDE \$1FDF | ROM | 4 \ | SERIAL COMMUNICATIONS STATUS REGISTER |
| | IRQ OPTION REG. | | SERIAL COMMUNICATIONS DATA REGISTER |
| Γ | BOOTSTRAP | | TIMER CONTROL REGISTER |
| \$1FEF | VECTORS (ROM) | 8175 256 BYTES | TIMER STATUS REGISTER |
| \$1FF0 | UNUSED | 8175 BYTES | INPUT CAPTURE HIGH REGISTER |
| \$1FF3 | 4 BYTES | 8179 | INPUT CAPTURE LOW REGISTER |
| \$1FF4 | USER VECTORS | 8180 | OUTPUT COMPARE HIGH REGISTER |
| excec | EEPROM 12 BYTES | | OUTPUT COMPARE LOW REGISTER |
| \$1FFFL | ere ver stigninger Tillsperate | L 8191 | COUNTER HIGH REGISTER |
| | | | COUNTER LOW REGISTER |
| | | | ALTERNATE COUNTER HIGH REGISTER |
| | | 1 | ALTERNATE COUNTER LOW REGISTER |
| | | 1 | PROGRAM REGISTER |
| | | i | BREAKPOINT ADDRESS LOW |
| | | | BREAKPOINT ADDRESS HIGH |
| | | | UNUSED |

PROGRAMMING MODEL INTERRUPT STACKING ORDER

PROGRAMMING MODEL



NOTE: Since the Stack Pointer decrements during pushes, the PCL is stacked firstorial stack of the stack is in the reverse order. Go to: www.freescale.com

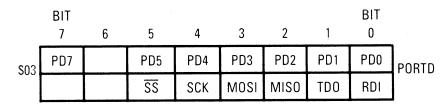
Freescale Semiconductor, Inc. REGISTER AND CONTROL BIT SUMMARY

| | BIT 7 | 6 | 5 | 4 | 3 | 2 | 1 | BIT | |
|------|----------|------|-------|-------|-------|-------|-------|----------|--------|
| \$00 | , PA7 | PA6 | PA5 | PA4 | PA3 | PA2 | PA1 | 0 PA0 | PORTA |
| \$01 | PB7 | PB6 | PB5 | PB4 | PB3 | PB2 | PB1 | PB0 | PORTB |
| \$02 | PC7 | PC6 | PC5 | PC4 | PC3 | PC2 | PC1 | PCO | PORT C |
| \$03 | PD7/* | 1.00 | PD5/* | PD4/* | PD3/* | PD2/* | PD1/* | PD0/* | PORTD |
| \$04 | DDA7 | DDA6 | DDA5 | DDA4 | DDA3 | DDA2 | DDA1 | DDA0 | DDRA |
| \$05 | DDB7 | DDB6 | DDB5 | DDB4 | DDB3 | DDB2 | DDB1 | DDB0 | DDRB |
| \$06 | DDC7 | DDC6 | DDC5 | DDC4 | DDC3 | DDC2 | DDC1 | DDC0 | DDRC |
| \$07 | DDD7 | 0000 | DDD5 | DDD4 | DDD3 | DDD2 | DDD1 | DDDO | |
| \$08 | 0007 | | 0003 | 0004 | 0003 | 0002 | 1000 | 0000 | UNUSED |
| \$09 | | | | | | | | | UNUSED |
| \$0A | SPIE | SPE | DWOM | MSTR | CPOL | СРНА | SPR1 | SPRO | SPCR |
| \$0B | SPIF | WCOL | | MODF | | | | | SPSR |
| \$0C | SPD7 | SPD6 | SPD5 | SPD4 | SPD3 | SPD2 | SPD1 | SPD0 | SPDR |
| \$0D | | | SCP1 | SCP0 | | SCR2 | SCR1 | SCRO | BAUD |
| \$0E | R8 | T8 | | М | WAKE | | | | SCCR1 |
| \$0F | TIE | TCIE | RIE | ILIE | TE | RE | RWU | SBK | SCCR2 |
| \$10 | TDRE | TC | RDRF | IDLE | OR | NF | FE | | SCSR |
| \$11 | SCD7 | SCD6 | SCD5 | SCD4 | SCD3 | SCD2 | SCD1 | SCDO | SCDAT |
| \$12 | ICIE | OCIE | TOIE | 0 | 0 | 0 | IEDG | OLVL | TCR |
| \$13 | ICF | OCF | TOF | 0 | 0 | 0 | 0 | 0 | TSR |
| \$14 | ICH7 | ICH6 | ICH5 | ICH4 | ICH3 | ICH2 | ICH1 | ICHO | ICHR |
| \$15 | ICL7 | ICL6 | ICL5 | ICL4 | ICL3 | ICL2 | ICL1 | ICLO | ICLR |
| \$16 | OCH7 | OCH6 | OCH5 | OCH4 | OCH3 | OCH2 | OCH1 | OCHO | OCHR |
| \$17 | OCL7 | OCL6 | OCL5 | OCL4 | OCL3 | OCL2 | OCL1 | OCLO | OCLR |
| \$18 | CH7 | CH6 | CH5 | CH4 | CH3 | CH2 | CH1 | CHO | CHR |
| \$19 | CL7 | CL6 | CL5 | CL4 | CL3 | CL2 | CL1 | CLO | CLR |
| \$1A | ACH7 | ACH6 | ACH5 | ACH4 | ACH3 | ACH2 | ACH1 | ACHO | ACHR |
| \$1B | ACL7 | ACL6 | ACL5 | ACL4 | ACL3 | ACL2 | ACL1 | ACLO | ACLR |
| \$1C | | | | | | | | | UNUSED |
| \$1D | | | | | | | | | UNUSED |
| \$1E | | | | | | | | | UNUSED |
| \$1F | | | | | | | | | UNUSED |

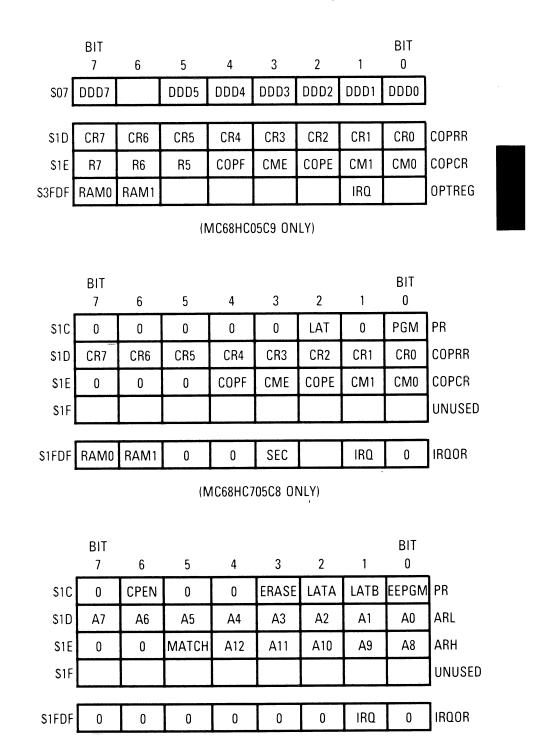
*Denotes fixed input port, see following page.

[†]MC68HC05C9 only For More Information On This Product, Go to: www.freescale.com

Freescale Semiconductor, Inc. REGISTER AND CONTROL BIT SUMMARY



(PORT D FIXED INPUT REGISTER)



ACHR Freescale Semiconductor, Inc.

| | Alter | nate (| Count | er Hig | h Reg | ister (| ACHR |) \$1A |
|----|-------|--------|-------|--------|-------|---------|------|--------|
| | 7. | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ACH7 | ACH6 | ACH5 | ACH4 | ACH3 | ACH2 | ACH1 | ACH0 |
| RI | ESET | | | | | | | |
| | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |

ACLR

| Alte | rnate | Count | er Lov | w Reg | ister (| |) \$1B |
|-------|-------|-------|--------|-------|---------|------|--------|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ACL7 | ACL6 | ACL5 | ACL4 | ACL3 | ACL2 | ACL1 | ACLO |
| RESET | | | | | | | |
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |

ARH

(MC68HC805C4 ONLY)

Hardware Breakpoint Register High (ARH) \$1E

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|-----------|---|-------|-----|-----|-----|----|----|--|
| | 0_ | 0 | МАТСН | A12 | A11 | A10 | A9 | A8 | |
| R | ESET 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

 $\mathsf{MATCH}-\mathsf{An}$ instruction with the same address as that in the breakpoint register was fetched.

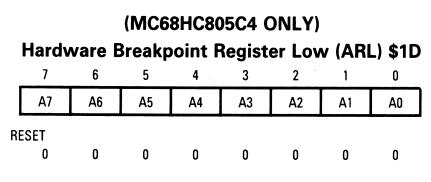
1 = Breakpoint enabled

0 = Breakpoint disabled

A12-A8 — Breakpoint address bits A12-A8

ARL

ARL



Freescale Semiconductor, Inc. BAUD

Baud Rate Register (BAUD) \$0D

The baud rate register is used to select the SCI transmitter and receiver baud rate. SCP0 and SCP1 prescaler bits are used in conjunction with the SCR2–SCR0 baud rates to provide multiple baud rate combinations for a given crystal freguency. Bits 3, 6, and 7 always read zero.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|----|------|---|------|------|---|------|------|------|--|
| | | — | SCP1 | SCPO | | SCR2 | SCR1 | SCR0 | |
| RI | ESET | | | | | | | | |

_ _ 0 0 _ U U

SCP0 — SCI Prescaler Bit 0

SCP1 — SCI Prescaler Bit 1

Two prescaler bits are used to increase the range of standard baud rates controlled by the SCR2–SCR0 bits. Prescaler internal processor clock division versus bits levels are listed in Table 1.

SCR0 — SCI Baud Rate Bit 0

SCR1 — SCI Baud Rate Bit 1

SCR2 — SCI Baud Rate Bit 2

Three baud rate bits are used to select the baud rates of the SCI transmitter and SCI receiver. Baud rates versus bit levels are listed in Table 2.

Tables 1 and 2 tabulate the divide chain used to obtain the baud rate clock (transmit clock). The actual divider chain is controlled by the combined SCP1–SCP0 and SCR2–SCR0 bits in the baud rate register. All divided frequencies shown in Table 1 represent the final baud rate resulting from the internal processor clock division shown in the divided by column only (prescaler division only). Table 2 lists the prescaler output divided by the action of the SCI select bits (SCR2–SCR0). For example, assume that a 9600 Hz baud rate is required with a 2.4576 MHz external crystal. In this case, the prescaler bits (SCP1–SCP0) could be configured as a divided-by-two. Using the same crystal, the 9600 baud rate can be obained with a prescaler divided-by-one and the SCR2–SCR0 bits configured for a divide-by-eight.

> For More Information On This Product, Go to: www.freescale.com

BAUD

| 0 |
|----------|
| |
| |
| |
| |
| |
| 0 |
| |
| 0 |
| |
| |
| T |
| C |
| |
| Q |
| U |
| |
| |
| |
| U |
| 5 |
| |
| 0 |
| |
| 1 |
| |
| Ö |
| 5 |
| 0 |
| ð |
| |
| 67.5 |
| |



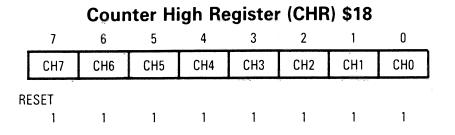
| B | | |) Fre | | e Ser | nio | | ndu | | to | ° r , |
|------------------------------------|------------------------------|------------|---|--|--|--|-------------|---------------------------|--------------------|------------|------------------------|
| | | 1.8432 | 57.60 kHz 19.20 kHz 14.40 kHz 4430 Hz | an be obtained b urther division us | | | 9600 Hz | 2000 Hz | 4000 Hz 2400 Hz | 1200 Hz | 600 Hz |
| | | 2.0 | 62.60 kHz 20.833 kHz 15.625 kHz 4800 Hz | ud rate (Tx) that c | Output | ut | 19.20 kHz | 19.20 kHz 9600 H- | 4800 Hz | 2400 Hz | 0 Hz |
| iency Output | :y MHz | 2.4576 | 76.80 kHz 25.60 kHz 19.20 kHz 5.907 kHz | nest transmit bar may be obtained | en Prescaler | Baud Rate Outp | 19.2 | 19.2 | 480 | 240 | 120 |
| Highest Baud Rate Frequency Output | Crystal Frequency MHz | 4.0 | 125.000 kHz 41.666 kHz 31.250 kHz 9600 Hz | t baud rates which are the highest transmit baud rate (Tx) that can be obtained by a lifer division. Lower baud rates may be obtained by providing a further division using presentative pr | Rate Output for a Given Prescaler Output | Representative Highest Prescaler Baud Rate Output | 76.80 kHz | 76.80 kHz 38 40 kHz | 19.20 kHz | 9600 Hz | 4800 Hz |
| | | 4.194304 | 131.072 kHz 13 43.691 kHz 332.768 kHz 10.082 kHz 10.082 kHz 10.082 kHz 10.082 kHz 10.082 kHz 10.082 kHz 10.082 kHz 10.081 | l represent baud rates the prescaler division. r some representative | 1 | Representative | 32.768 kHz | | | | |
| Table 1. Prescaler | | 8.0 | 250.00 KHz 83.332 KHz 62.500 KHz 19.200 KHz | Refers to the internal processor clock. NOTE: The divided frequencies shown in Table 1 represent baud rates which are the highest transmit baud rate (Tx) that can be obtained by a specific crystal frequency and only using the prescaler division. Lower baud rates may be obtained by providing a further division using the SCI rate select bits as shown below for some representative prescaler outputs. | Table 2. Transmit Baud | | 131.072 kHz | 131.072 kHz 65.536 kHz | 32.768 kHz | 16.384 kHz | 0.132 KHZ 4 096 kHz |
| | Clock* | Divided By | 134 00 - | A Refers to the internal processor clock. Do NOTE: The divided frequencies shown specific crystal frequency and o the SCI rate select bits as shown | | Divided By ded | | | | م م | 30 |
| | SCP Bit | 0 | 0-0- | The d specified | | | 0 | 0- | 0, | - 0 | > < |
| | SCI | - | 00 | Refe | | SCR Bits | 2 | 00 | | |) C |
| B | 4 U | ID | Fo | r More In | nforma to: ww | tion w.f | n O ree | n Th sca | is le.e | Pro | od m |

| N N | SCR Bits | Bits | Divided | l able Z. I ransmit | _ | uu nate Output TOF a GIVEN Prescaler U Representative Highest Prescaler Baud Rate Output | Baud hate Output Tor a GIVEN Prescaler Output Representative Highest Prescaler Baud Rate Output | |
|-----|----------|---------------------|--|--|--|---|--|---|
| ~ | - | 0 | Βγ | 131.072 kHz | 32.768 kHz | 76.80 kHz | 19.20 kHz | 9600 Hz |
| 0 | 0 | 0 | - | 131.072 kHz | 32.768 kHz | 76.80 kHz | 19 20 kHz | |
| 0 (| 0 | - (| 7 | 65.536 kHz | 16.384 kHz | 38.40 kHz | 9600 Hz | 2000 Hz |
| 0 | | 0 | 4 | 32.768 kHz | 8.192 kHz | 19.20 kHz | 4800 Hz | |
| | | | ∞ | 16.384 kHz | 4.096 kHz | 9600 Hz | 2400 H2 | 1200 12 |
| | 0 | 0 | 16 | 8.192 kHz | 2.048 kHz | 4800 Hz | 1200 H2 | |
| | 0 | <u></u> | 32 | 4.096 kHz | 1.024 kHz | 2400 Hz | 200 H2 | 300 H2 |
| | | 0 | 64 | 2.048 kHz | 512 Hz | 1200 Hz | 300 Hz | 150 Hz |
| | - | - | 128 | 1.024 kHz | 256 Hz | 600 Hz | 150 Hz | 75 Hz |
| Z | Ë | Tabl freq and | le 2 illust uency. Th the receiv | NOTE: Table 2 illustrates how the SCI select bits can be used to provide tower transmitter baud rate by further dividing the prescaler output frequency. The five examples are only representative samples. In all cases, the baud rates shown are transmit baud rates (transmit clock) and the receive clock is 16 times higher in frequency than the actual baud rate. | bits can be used to provide lower trans representative samples. In all cases, the er in frequency than the actual baud rate. | vide tower transmitter t In all cases, the baud ra actual baud rate. | be used to provide tower transmitter baud rate by further dividing the prescaler output native samples. In all cases, the baud rates shown are transmit baud rates (transmit clock) uency than the actual baud rate. | ting the prescaler output aud rates (transmit clock) |
| | | | | | | זכומתו המתח ומוכי | | |

Freescale Semiconductor, Inc.

Downloaded from Elcodis.com electronic components distributor

Freescale Semiconductor, Inc. CHR



CLR

CLR

| | | Cour | nter Lo | ow Re | gister | · (CLR |) \$19 | |
|----|-----------|------|---------|-------|--------|--------|--------|-----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | CL7 | CL6 | CL5 | CL4 | CL3 | CL2 | CL1 | CLO |
| RE | ESET 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |

Freescale Semiconductor, Inc.

Freescale Semiconductor, Inc. COPCR

(MC68HC705C8 ONLY)

COP Control Register (COPCR) \$1E

The COPCR shown below is used to control the COP watchdog timer and clock monitor functions.

| 7 6 5 4 3 0 0 0 COPF CN | | 1 CM1 | | | | | | | | |
|--|-----------|----------|----------|--|--|--|--|--|--|--|
| 0 0 0 COPF CN | E COPE | I CM1 | | | | | | | | |
| | | | CM0 | | | | | | | |
| RESET | | | | | | | | | | |
| 0 0 0 0 0 | 0 | 0 | 0 | | | | | | | |
| COPF – Computer Operating Proper | L | | | | | | | | | |
| 1 = COP or clock monitor reset has | • | d | | | | | | | | |
| 0 = No COP or clock monitor rese | t has occ | urred | | | | | | | | |
| CME — Clock Monitor Enable | | | | | | | | | | |
| 1 = Clock monitor enabled | | | | | | | | | | |
| 0 = Clock monitor disabled | . – | | | | | | | | | |
| COPE — Computer Operating Properly Enable | | | | | | | | | | |
| 1 = COP timeout enabled | | | | | | | | | | |
| 0 = COP timeout disabled CM1 — Computer Operating Pro | orby Mo | do 1 | | | | | | | | |
| Used in conjunction with C | | | h tha C | | | | | | | |
| timeout period. CM1 can be | | | | | | | | | | |
| is cleared only by reset. See | | i set an | iyune, | | | | | | | |
| CM0 — Computer Operating Pro | | de 0 | | | | | | | | |
| Used in conjunction with CN | | | | | | | | | | |
| timeout period. CM0 can be | | | | | | | | | | |
| cleared only by reset. See Ta | | Julian | ytime, t | | | | | | | |
| Bits 7–5 — Not used* | | | | | | | | | | |
| Always read zero | | | | | | | | | | |

*In the MC68HC05C9, these bits (R7–R5) are reserved factory test bits.

| ore | | | | | | | |
|---------|-----|-----|---------------------------------|--------------------------------|--------------------------------------|--------------------------------|--------------------------------|
| e Infoi | CM1 | CM0 | E/2 ¹⁵ Divided By | XTAL = 4.0 MHz, E = 2.0 MHz | XTAL = 3.5795 MHz, E = 1.7897 MHz | XTAL = 2.0 MHz, E = 1.0 MHz | XTAL = 1.0 MHz, E = 0.5 MHz |
| mat | 0 | 0 | - | 16.38 ms | 18.31 ms | 32.77 ms | 65.54 ms |
| tion | 0 | - | 4 | 65.54 ms | 73.24 ms | 131.07 ms | 262.14 ms |
| On | - | 0 | 16 | 262.95 ms | 292.95 ms | 524.29 ms | 1.048 s |
| This | - | - | 64 | 1.048 s | 1.172 s | 2.097 s | 4.194 s |
| s P | | | | | | | |

(MC68HC705C8 AND MC68HC05C9 ONLY)

Table 3. COP Timeout Period

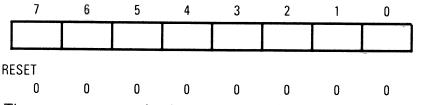
Г

Freescale Semiconductor, Inc. COPCR

COPRR Freescale Semiconductor, Inc.

COP Reset Register (COPRR) \$1D

The COPRR shown below is used to control the COP watchdog timer and clock monitor functions.



The sequence required to reset the COP timer is as follows: Write \$55 to the COP reset register.

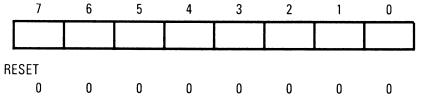
Write \$AA to the COP reset register.

Both write operations must occur in the order listed, but any number of instructions may be executed between the two write operations. The elapsed time between software resets must not be greater than the COP timeout period. Reading the COP reset register does not return valid data and does not affect the watchdog timer.

(MC68HC705C8 AND MC68HC05C9 ONLY)

COP Reset Register (COPRR) \$1D

The COPRR shown below is used to control the COP watchdog timer and clock monitor functions.



The sequence required to reset the COP time is as follows: Write \$55 to the COP reset register.

Write \$AA to the COP reset register.

Both write operations must occur in the order listed, but any number of instructions may be executed between the two write operations. The elapsed time between software resets must not be greater than the COP timeout period. Reading the COP reset register does not return valid data and does not affect the watchdog timer.

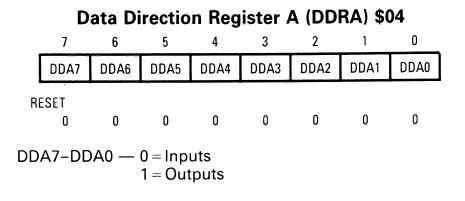
(MC68HC705C8 AND MC68HC05C9 ONLY)

For More Information On This Product,

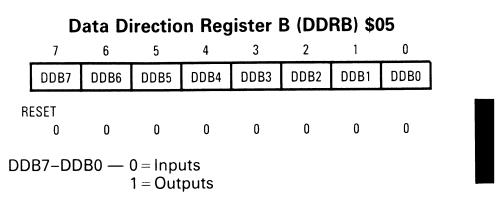
Go to: www.freescale.com

COPRR

Freescale Semiconductor, Inc. DDRA

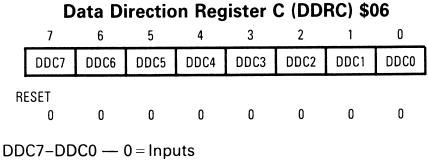


DDRB



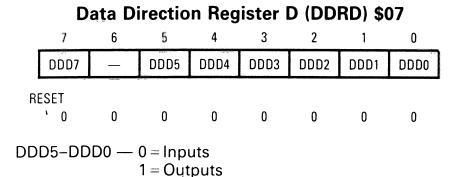
DDRC

DDRC



1 = Outputs

Freescale Semiconductor, Inc.



Bits 7,6 — Not used.

ICHR

| | Inj | put Ca | pture | High | Regis | ter (IC | :HR) \$ | 514 |
|----|----------|--------|-------|------|-------|---------|---------|------|
| _ | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ICH7 | ICH6 | ICH5 | ICH4 | ICH3 | ICH2 | ICH1 | ICHO |
| RE | SET U | U | U | U | U | U | U | U |

ICLR

| | In | put Ca | apture | e Low | Regis | ter (IC | :LR) \$ | 15 |
|----|-----------|--------|--------|-------|-------|---------|---------|--------|
| _ | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | ICL7 | ICL6 | ICL5 | ICL4 | ICL3 | ICL2 | ICL1 | ICLO |
| RE | ESET U | U | U | U | U | U | U | ۰ U |

Downloaded from Elcodis.com electronic components distributor

Freescale Semiconductor, Inc. IRQOR

(MC68HC705C8 ONLY)

Option Register (IRQOR) \$1FDF

The option register is used to select the \overline{IRQ} sensitivity, enable the PROM security, and select the memory configuration.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | - |
|----|------|------|---|---|-----|---|-----|---|---|
| | RAM0 | RAM1 | 0 | 0 | SEC | _ | IRQ | 0 | |
| RI | ESET | 0 | 0 | Λ | 11 | | 1 | 0 | |

RAM0-Random Access Memory Control Bit 0

1 = Maps 32 bytes of RAM into page zero starting at address \$0030. Addresses from \$0020 to \$0030 are reserved. This replaces 48 bytes of PROM that were used at these locations. This bit can be read or written at any time, allowing memory configuration to be changed during program execution.

0 = Provides 48 bytes of PROM at location \$0030.

RAM1-Random Access Memory Control Bit 1

1 = Maps 96 bytes of RAM into page zero starting at address \$0100. This replaces 96 bytes of PROM that were used at these locations. This bit can be read or written at any time, allowing memory configuraton to be changed during program execution.

0 = Provides 96 bytes of PROM at location \$0100.

SEC – Security

1 = Bootloader disabled, MCU operates only in single-chip mode.

0 = Security off, bootloader enabled, expanded mode enabled.

- IRQ-Interrupt Request Bit Sensitivity
 - $1 = \overline{IRQ}$ pin is both negative edge- and level-sensitive.

 $0 = \overline{IRQ}$ pin is negative edge-sensitive only.

IRO is set only by reset, but can be cleared by software.

This can only be written once.

- Bit 0, 4, 5
 - Always read zero.
- Bit 2

Can be either one or zero.

For More Information On This Product, Go to: www.freescale.com

IRQOR

Freescale Semiconductor, Inc. IRQOR

(MC68HC05C9 ONLY)

Option Register (IRQOR) \$3FDF

The option register is used to select the IRO sensitivity, enable the ROM security, and select the memory configuration.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|------|------|---|---|---|---|-----|---|
| | RAM0 | RAM1 | 0 | 0 | 0 | 0 | IRQ | 0 |
| RI | ESET | | | | | | | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

RAM0-Random Access Memory Control Bit 0

1 = Maps 32 bytes of RAM into page zero starting at address \$0020. This replaces 48 bytes of ROM that were used at these locations. This bit can be read or written at any time, allowing memory configuration to be changed during program execution.

0 = Provides 48 bytes of ROM at location \$0020.

RAM1-Random Access Memory Control Bit 1

1 = Maps 128 bytes of RAM into page zero starting at address \$0100. This replaces 128 bytes of ROM that were used at these locations. This bit can be read or written at any time, allowing memory configuraton to be changed during program execution.

0=Provides 128 bytes of ROM at location \$0100.

IRQ-Interrupt Request Bit Sensitivity

 $1 = \overline{IRQ}$ pin is both negative edge- and level-sensitive.

 $0 = \overline{IRO}$ pin is negative edge-sensitive only.

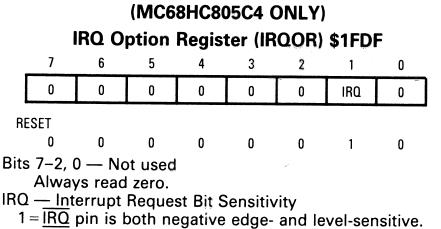
IRQ is set only by reset, but can be cleared by software.

This can only be written once.

Bit 0, 2, 3, 4, 5

Always read zero.

Freescale Semiconductor, Inc. IRQOR



 $0 = \overline{IRO}$ pin is negative edge-sensitive only.

IRQ is set only by reset, but can be cleared by software.

OCHR

Output Compare High Register (OCHR) \$16

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|------|------|------|------|------|------|------|------|
| | OCH7 | OCH6 | OCH5 | OCH4 | OCH3 | OCH2 | OCH1 | ОСНО |
| RE | ESET | | 11 | | • • | | | |

OCLR

| | Out | put Co | ompai | re Low | v Regi | ster (| OCLR) | \$17 |
|-------|------|--------|-------|--------|--------|--------|-------|------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | OCL7 | OCL6 | OCL5 | OCL4 | OCL3 | OCL2 | OCL1 | OCLO |
| RESET | | | | | | | | |
| | U | U | U | U | U | U | U | U |

Freescale Semiconductor, Inc. PORTA

| | | Port / | A Data | a Regi | ster (F | ORIA | 4) \$00 | |
|---|-----------|--------|--------|--------|---------|------|---------|-----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PA7 | PA6 | PA5 | PA4 | PA3 | PA2 | PA1 | PA0 |
| R | ESET U | U | U | U | U | U | U | U |

.....

PORTB

| | | Port I | B Data | n Regi | ster (F | PORTE | B) \$01 | |
|---|-----------|--------|--------|--------|---------|-------|---------|-----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PB7 | PB6 | PB5 | PB4 | PB3 | PB2 | PB1 | PB0 |
| R | ESET U | U | U | U | U | U | U | U |

PORTC

Port C Data Register (PORTC) \$02

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|----------|-----|-----|-----|-----|-----|-----|-----|
| | PC7 | PC6 | PC5 | PC4 | PC3 | PC2 | PC1 | PCO |
| RE | SET U | U | U | U | U | U | U | U |

PORTD

| | | Port I | D Data | n Regi | ster (F | PORTE |) \$03 | |
|----|------------|--------|--------------------------------|----------------------|-------------------------|-------------------------|------------------|-------------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PD7 | | PD5 | PD4 | PD3 | PD2 | PD1 | PD0 |
| RE | SET PD7 | 0 | P <u>D5</u> / SS (Port D | PD4/ SCK fixed | PD3/ MOSI input r | PD2/ MISO egister | PD1/ TD0) | PD0/ RDI |
| RE | SET U | U | U | U | U | U | U | U |

(MC68HC705C8 ONLY)

Program Register (PR) \$1C

The program register (\$1C) is used to perform PROM programming.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|------|---|---|---|---|-----|---|-----|
| | 0 | 0 | 0 | 0 | 0 | LAT | 0 | PGM |
| R | ESET | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

LAT — Latch Enable

- 1 = Enables PROM data and address bus latches for programming or erasing on the next byte write cycle.
- 0 = Latch disabled. PROM data and address buses are unlatched for normal CPU operations.

This bit is both readable and writable.

PGM — Program

1 =Applies V_{PP} power to the PROM for programming.

- $0 = V_{PP}$ power off.
- If LAT is cleared, PGM cannot be set.

Bits 1, 7-3 — Not Used

Always read zero.

(MC68HC805C4 ONLY)

Program Register (PR) \$1C

The program register (\$1C) is used for single-byte EEPROM programming.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|----|------|---|---|-------|------|------|-------|
| Γ | 0 | CPEN | 0 | 0 | ERASE | LATA | LATB | EEPGM |
| | 14 | | | | | | | |

RESET

0 0 0 0 0 0 0

CPEN — Charge Pump Enable

1 = Charge pump enabled

0 = Charge pump disabled

ERASE — Erase EEPROM Enable

- 1 = Erase enabled
- 0 = Erase disabled

LATA — Latch A Enable

- 1 = Enables array A data and address bus latches for programming or erasing on the next byte write cycle.
- 0 = Latch disabled
- LATB Latch B Enable
 - 1 = Enables array B data and address bus latches for programming or erasing on the next byte write cycle.
 - 0 = Latch disabled
 - Note: If LATA and LATB are cleared, EEPGM cannot be set.

PR Freescale Semiconductor, Inc.

EEPGM — Electrically Erase/Program

1 = Applies Vpp power to the EEPROM array for programming or erasing operation.

 $0 = V_{PP}$ power off

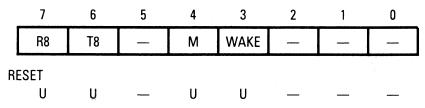
Bits 4, 5, 7 — Not used

Always read zero.

SCCR1

Serial Communications Control Reg. 1 (SCCR1) \$0E

The SCCR1 register control bits determine word length and select the wake-up method.



R8 — Receive Data Bit 8

R8 bit provides storage location for the ninth bit in the receive data byte (if M = 1).

T8 — Transmit Data Bit 8

T8 bit provides storage location for the ninth bit in the transmit data byte (if M = 1).

M — SCI Character Word Length

1 = One start bit, nine data bits, one stop bit

0 =One start bit, eight data bits, one stop bit

WAKE — Wake-Up Select

Wake bit selects the receiver wake-up method.

1 = Address bit (most significant bit)

0 = Idle line condition

Bits 2–0 and 5 — Not used

Can read either one or zero.

The address bit is dependent on both the wake-bit and the m-bit level. Additionally, the receiver does not use the wake-up feature unless the RWU control bit is SCCR2 is set.

| Wake | М | Receiver Wake-Up |
|------|---|---|
| 0 | Х | Detection of an idle line allows the next data byte received to cause the receive data register to fill and produce an RDRF flag. |
| 1 | 0 | Detection of a received one in the eighth data bit allows an RDRF flag and associated error flags. |
| 1 | 1 | Detection of a received one in the ninth data bit allows an RDRF flag and associated error flags. |

For More Information On This Product, Go to: www.freescale.com

SCCR1

Serial Communications Control Reg. 2 (SCCR2) \$0F

The SCCR2 provides control of individual SCI functions such as interrupts, transmit/receive enabling, receiver wakeup, and break code.

| NOC | | | | | | | | |
|-----|---|----------|----------|---------|---------|---------|--------|--------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | TIE | TCIE | RIE | ILIE | TE | RE | RWU | SBK |
| Rí | ESET | | | | | | | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| TIĘ | — Tra | insmit | Interru | pt Ena | ble | | | |
| 1 | ≔ SCI | interru | pt enal | bled | | | | |
| | 0 = TDRE interrupt disabled | | | | | | | |
| | TCIE — Transmit Complete Interrupt Enable | | | | | | | |
| | | interru | | | | | | |
| | | nterrup | | | | | | |
| | | ceive lı | | | le | | | |
| | | interru | • | | | | | |
| | | Fand | | | | ed | | |
| | | e Line | | • | ble | | | |
| | | interru | | | | | | |
| | | interru | • | bled | | | | |
| | | smit Ena | | | | | | |
| 1 | = Tran | smit sh | ift regi | ster ou | tput is | applied | to the | TDO li |

- = Transmit shift register output is applied to the TDO line. Depending upon the SCCR1 M bit, a preamble of 10 (M = 0) or 11 (M = 1) consecutive ones is transmitted.
- 0 = Transmitter disabled after last byte is loaded in the SCDAT and TDRE is set. After last byte is transmitted, TDO line becomes a high-impedance line.
- RE Receive Enable
 - 1 = Receiver shift register input is applied to the RDI line.
 - 0 = Receiver disabled and RDRF, IDLE, OR, NF, and FE status bits are inhibited.

SCCR2 Freescale Semiconductor, Inc.

- RWU Receiver Wake-Up
 - 1 == Places receiver in sleep mode and enables wake-up function.
 - 0 = Wake-up function disabled after receiving data word with MSB set (if WAKE = 1). Wake-up function also disabled after receiving 10 (M = 0) or 11 (M = 1) consecutive ones (if WAKE=0).
- SBK Send Break
 - 1 = Transmitter continually sends blocks of zeros (sets of 10 or 11) until cleared. Upon completion of break code, transmitter sends one high bit for recognition of valid start bit.
 - 0 = Transmitter sends 10 (M = 0) or 11 (M = 1) zeros then reverts to an idle state or continues sending data. If transmitter is empty and idle, setting and clearing the SBK bit may queue up to two character times of break because the first break transfer immediately to the shift register, and the second is queued into the parallel transmit buffer.

SCDAT

Serial Communications Data Register (SCDAT) \$11

The SCDAT is a read/write register used to receive and transmit SCI data.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|------|------|------|------|------|------|------|------|
| | SCD7 | SCD6 | SCD5 | SCD4 | SCD3 | SCD2 | SCD1 | SCD0 |
| RE | SET | 11 | 11 | | 11 | | 11 | |

SCSR

Serial Communications Status Register (SCSR) \$10

The SCSR provides inputs to the SCI interrupt logic circuits. Noise flag and framing error bits are also contained in the SCSR.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|----|-----------|-----|------|------|----|----|----|---|---|
| | TDRE | TC6 | RDRF | IDLE | OR | NF | FE | _ | |
| RE | ESET 1 | 1 | 0 | 0 | 0 | 0 | 0 | | • |

TDRE — Transmit Data Register (TDR) Empty

- 0 = TDR contents transferred to the transmit data shift register.
- 0 = TDR still contains data. TDRE is cleared by reading the SCSR followed by a write to the TDR.

SCSR

Freescale Semiconductor, Inc. SCSR

- TC Transmit Complete
 - 1 = Indicates end of data frame, preamble, or break condition has occurred.
 - 0 = TC bit cleared by reading the SCSR, followed by a write to the TDR.
- RDRF-Receive Data Register (RDR) Full
 - 1 = Receive data shift register contents transferred to the RDR.
 - 0 = Receive data shift register transfer did not occur. RDRF is cleared by reading the SCSR, followed by a read of the RDR.
- IDLE Idle Line Detect
 - 1 = Indicates receiver has detected an idle line.
 - 0=IDLE is cleared by reading the SCSR, followed by a read of the RDR. Once IDLE is cleared, IDLE cannot be set until RDI line becomes active and idle again.
- OR Overrun Error
 - 1 = Indicates receive data shift register data is sent to a full RDR (RDRF = 1). Data causing the overrun is lost, and RDR data is not disturbed.
 - 0 = OR is cleared by reading the SCSR, followed by a read of the RDR.
- NF Noise Flag
 - 1 = Indicates noise is present on the receive bits, including the start and stop bits. NF is not set until RDRF=1.
 - 0 = NF is cleared by reading the SCSR, followed by a read of the RDR.
- FE Framing Error
 - 1 = Indicates stop bit not detected in received data character. FE is set the same time RDRF is set. If received byte causes both framing and overrun errors, processor will only recognize the overrun error. Further data transfer into the RDR is inhibited until FE is cleared.
 - 0 = FE is cleared by reading the SCSR, followed by a read of the RDR.
- Bit 0 Not used
 - Can read either one or zero.

SPCR Freescale Semiconductor, Inc.

Serial Peripheral Control Register (SPCR) \$0A

The SPCR provides control of individual SPI functions such as interrupt and system enabling disabling, master slave mode select, and clock polarity phase rate select.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|------|--|----------|---------------------|----------|---------|--------|-----------|-----------|------|
| 9 | SPIE | SPE | DW0M* | MSTR | CPOL | СРНА | SPR1 | SPRO | 1 |
| RESE | ET | | | | | | | | |
| | 0 | 0 | | 0 | U | U | U | U | |
| SPIE | — Se | erial P | eripher | al Inter | rupt E | nable | | | |
| | | | rupt en | | | | | | |
| | | | rupt di | | _ | | | | |
| | | | eriphera | il Syste | em Ena | able | | | |
| | | Pl syste | em on em off | | | | | | |
| | | | | Select | | | | | |
| | MSTR — Master Mode Select 1 = Master mode | | | | | | | | |
| 0 | = Sla | ave m | ode | | | | | | |
| CPOL | . — (| Clock F | Polarity | | | | | | |
| | • | • | bit cor | | | | | is used | d in |
| | | | ith the | | ohase (| (CPHA) | bit. | | |
| | | | idles h idles i | | toto | | | | |
| | | Clock I | | | siale | | | | |
| | | | bit alon | a with | CPOL | contro | ls the | clock-c | data |
| | | | etween | | | | | | |
| sel | ect <u>s</u> | one or | ^r two cl | ocking | protoc | cols. | | | |
| | | | output | | | | - <u></u> | | |
| 0 | | | <u>c</u> k is th | | | | | c | • . |
| | W | hen S | S is low | i, first | edge o | t SCK | Invoke | s first c | data |

sample.

SPR1-SPR0 — SPI Clock Rate Bits

Two clock rate bits are used to select one of four clock rates to be used as SCK in the master mode. In the slave mode, the two clock rate bits have no effect. Clock rate selection is shown in the following table.

| SPR1 | SPR0 | Internal Processor Clock Divided By |
|------|------|--|
| 0 | 0 | 2 |
| 0 | 1 | 4 |
| 1 | 0 | 16 |
| 1 | 1 | 32 |

Bit 5 — Not used*

SPCR

Can read either one or zero.

(*MC68HC05C9 only, bit 5 (DWOM) is the wire-OR mode bit.)

- 1 = Disables active pullup devices on Port D, causing outputs to be open drain.
- 0 = Open-drain disabled.

Serial Peripheral Data I/O Register (SPDR) \$0C

The SPDR is read/write register used to receive and transmit SPI data.

| _ | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|----|-----------|------|------|------|------|------|------|------|--|
| | SPD7 | SPD6 | SPD5 | SPD4 | SPD3 | SPD2 | SPD1 | SPD0 | |
| RE | ESET U | U | U | U | U | U | U | U | |

A write to the SPDR places data directly into the shift register for transmission. Only a write to this register will initiate transmission/reception of another byte and will only occur in the master device. On completion of byte transmission, the SPIF status bit is set in both master and slave devices.

A read to the SPDR causes the buffer to be read. The first SPIF status bit must be cleared by the time a second data transfer from the shift register to the read buffer begins, or an overrun condition will exist. In overrun cases, the byte causing the overrun is lost.

SPSR

SPSR

Serial Peripheral Status Register (SPSR) \$0B

The SPSR contains three status bits.

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|------|---|------|---|---|---|---|
| SPIF | WCOL | _ | MODF | — | | _ | — |

RESET

0 0 - 0 - - -

SPIF — Serial Peripheral Data Transfer Flag

- 1 = Indicates data transfer completed between processor and external device.
 - (If SPIF = 1 and SPIE = 1, SPI interrupt is enabled.)
- 0 = Clearing is accomplished by reading SPSR, followed by SPDR access.

WCOL — Write Collision

- 1 = Indicates an attempt is made to write to SPDR while data transfer is in processor.
- 0 = Clearing is accomplished by reading SPSR, followed by SPDR access.

MODF — Mode Fault Flag

- 1 = Indicates multi-master system control conflict.
- 0 = Clearing is accomplished by reading SPSR, followed by a write to the SPCR.

Bits 3–0 and 5 — Not used

Can read either zero or one.

For More Information On This Product, Go to: www.freescale.com

Timer Control Register (TCR) \$12

The TCR is a read/write register containing five control bits. Three bits control interrupts associated with the timer status register flags ICF, OCF, and TOF.

| 109 | | ug5 101 | , 001, | and i | 01. | | | | |
|----------|---------|---------|----------|----------|---------|---------|----------|----------|------|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| | ICIE | OCIE | TOIE | 0 | 0 | 0 | IEDG | OLVL | |
| RE | ESET | | | | | | | | |
| | 0 | 0 | 0 | 0 | 0 | 0 | U | 0 | |
| ICIE | E — Inp | out Cap | oture Ir | nterrup | t Enab | le | | | |
| | 1 = lnt | terrupt | enable | d | | | | | |
| | | | disabl | | | | | | |
| OC | | | Compa | | rrupt E | nable | | | |
| | | • | enable | | | | | | |
| TO | | | disabl | | _ | | | | |
| 101 | | | verflov | | rupt En | able | | | |
| | | • | enable | | | | | | |
| 150 | | • | disabl | ed | | | | | |
| | | put Ec | | | • | | | •.• | |
| | | | | | | | | nsition | |
| | | | | | unning | counte | er trans | sfer to | the |
| Ir | • | • | registe | r. | | | | | |
| | | sitive | 0 | | | | | | |
| <u> </u> | | egative | 0 | | | | | | |
| | | Output | | | | | | | |
| | | | | | | | | el regis | |
| h | v the n | ext su | ccessfi | il outri | it com | nare ar | nd will | annear | ' nn |

Value of output level is clocked into output level register by the next successful output compare and will appear on the TCMP pin.

- 1 = High output
- 0 = Low output
- Bits 2, 3, and 4 Not used Always read zero.



Timer Status Register (TSR) \$13

The TSR is a read-only register containing three status flag bits.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | _ |
|----|--------|--------|----------|---------|---------|--------|---------|---------|----|
| | ICF | OCF | TOF | 0 | 0 | 0 | 0 | 0 | |
| RI | ESET | | | | | | | | - |
| | U | U | U | 0 | 0 | 0 | 0 | 0 | |
| | — Inp | - | | - | | | | | |
| 1 | - Flag | cot wh | non coli | octod r | olarity | anha i | s sense | nd hv i | nn |

- 1 = Flag set when selected polarity edge is sensed by input capture edge detector.
- 0 = Flag cleared when TSR and input capture low register (\$15) are accessed.
- OCF Output Compare Flag
 - 1 = Flag set when output compare register contents match the free-running counter contents.
 - 0 = Flag cleared when TSR and output compare low register (\$17) are accessed.
- TOF Timer Overflow Flag
 - 1 = Flag set when free-running counter transition from \$FFFF to \$0000 occurs.
 - 0 = Flag cleared when TSR and counter low register (\$19) are accessed.
- Bits 4–0 Not used

Always read zero.

TSR

Freescale Semiconductor, Inc. ADDRESSING MODES

IMMEDIATE (IMM)

The effective address (EA) of an immediate mode instruction is the location following the opcode. This mode is used to fetch a value which is known at the time the program is written, and which is not changed during program execution.

DIRECT (DIR)

The EA of a direct mode instruction is the contents of the byte following the opcode. This mode is used to fetch a value from any one of the first 256 memory locations with a twobyte instruction.

EXTENDED (EXT)

The EA of an extended mode instruction is the contents of the next two bytes following the opcode. This mode is used to fetch a value from any location in the MC146805G2 memory location, I/O, RAM, and ROM, with a three-byte instruction.

INDEXED (IX, IX1, IX2)

The EA of an indexed mode instruction is determined by the contents of the X-register being added to an offset. The offset can be either zero, 8-bit, or 16-bit. For zero offset (IX), the X-register is the EA. For 8-bit offset (IX1), the result of the X-register contents added to the byte following the opcode is the EA. For 16-bit offset (IX2), the result of the Xregister contents added to the concatenated contents of the two bytes following the opcode is the EA.

RELATIVE (REL)

The EA of a relative mode instruction depends upon whether or not the branch is taken. If a branch is taken, EA is formed by adding the byte following the opcode to the value of the program counter, and the program counter is loaded with the EA. If no branch is required, EA is equal to the contents of the program counter.

BIT SET/CLEAR (BSC)

The EA of a Bit Set/Clear mode instruction is contained in the byte following the opcode. The actual bit which is to be set or cleared is contained in the lower four bits (nibble) of the opcode.

BIT TEST AND BRANCH (BTB)

This addressing mode combines direct, relative and bit addressing. The EA of this instruction is the contents of the byte following the opcode (direct mode), if no branch is taken. If a branch is taken, the EA becomes the result of the second byte following the opcode being added to the value of the program counter (similar to relative mode). The actual bit which is to be tested is contained in the lower four bits (nibble) of the opcode.

INHERENT (INH)

This addressing mode has no EA since all information necessary to carry out the instruction is contained in the opcode.

M68AC05 INSTRUCTION SET

The following table is an alphabetical listing of the instructions available to the M68HC05 MCU user. In listing all the factors necessary to program, the table uses the following symbols:

Condition Code Symblols

- H Half Carry (Bit 4)
- I Interrupt Mask (Bit 3)
- N Negate (Sign Bit 2)
- Z Zero (Bit 1)
- C Carry/Borrow (Bit 0)
- 💼 Test and Set if True,
 - (Cleared otherwise)
 - — Not Affected
- ? Load CC Reg. from Stack
- 0 Cleared
- 1 Set

Boolean Operators

| () | — Contents of (i.e.) (M) = | + | — (inclusive) OR |
|----|--|----------|------------------------------------|
| | means the contents | \oplus | - EXCLUSIVE OR |
| | of memory location | | — NOT |
| | Μ | - | — negation |
| | — is loaded with, 'gets' | | (twos complement) |
| • | — AND | × | multiplication |

MPU Registers

| A — Accumulator | PC | — Program Counter |
|--------------------------|-----|--------------------------------------|
| ACCA — Accumulator | PCH | — PC High Byte |
| CC — Condition Code Reg. | PCL | - PC Low Byte |
| X — Index Register | SP | — Stack Pointer |
| M — Any memory location | REL | Relative Address |
| (one byte) | | |

| Addressing Modes | (Abbreviation) | Opera | nds |
|--------------------|----------------|-------|-----|
| Inherent | INH | none | |
| Immediate | IMM | ii | |
| Direct (for bit | DIR | dd | |
| test instructions) | | dd | rr |
| Extended | EXT | hh | H |
| Indexed 0 Offset | IX | none | |
| Indexed 1-Byte | IX1 | ff | |
| Indexed 2-Byte | IX2 | ee | ff |
| Relative | REL | rr | |

INSTRUCTIONS, ADDRESSING MODES, AND EXECUTION TIMES

| | | ree | S | Cá | al | e . | Se | | | or | h | uq | ctq | pr, | _1 | ıc | - - | | |
|---------------------------------|------------|-------------------|-----|---------|------------|------------|-----|---------------|----------|-----|-----|-----|------|-----------------|-----|-----|------------|-----|----|
| de | ပ | \$ | | | | | | 41 |) | | | | | | | | | | |
| C C C | Z | \$ | | | | | | 41 | } | · | | | | 4 | > | | | | |
| lition | Z | 4• | | ····· | , | * | | 41 |) | | | | | \$1 | • | | | | |
| Condition Code | - | | | | | | | | | | | | | | | | | | |
| | I | \$ | | | | | | (1) | • | | | | | | | | | | |
| Bytes Cycles | | 2 | ო | 4 | 2 | 4 | m | 2 | ო | 4 | 5 | 4 | 3 | 2 | ო | 4 | വ | 4 | ო |
| Bytes | | 2 | 2 | ო | ო | 7 | - | 2 | 2 | ო | ო | 2 | 1 | 2 | 2 | n | e | 2 | - |
| ding 1aľ) | Operand | | | = | Ħ | | | | | = | Ħ | | | | | = | Ħ | | |
| Aachine Codin (hexadecimal) | | := ' | pp | ЧЧ | ee | Ħ | | := | pp | ЧЧ | ee | Ħ | | := | qq | hh | ee | Ħ | |
| Machine Coding (hexadecimal) | Opcode | A9 | 68 | 60 C | 6 0 | E9 | F9 | AB | BB | CB | DB | EB | FB | A4 | B4 | C4 | D4 | E4 | F4 |
| Addressing Mode for | Operand | MM | DIR | EXT | IX2 | IX1 | × | MMI | DIR | EXT | IX2 | IX1 | X | ININ | DIR | EXT | IX2 | IX1 | × |
| Boolean | Expression | ACCA ACCA + M + C | | | | | | ACCA ACCA + M | | | | | | ACCA & ACCA • M | | | | | |
| Operation | | Add with Carry | | | | | | Add | | | | | | Logical AND | | | | | |
| Source Form(c) | | den (opr) | M | ore | e li | nfo | orm | DD (opr) | on | Or | n T | his | ; Pi | SND (opr) | uci | t, | | | |

| ↓ ↓ | | | onductor, Inc. | 1 | |
|--|--|-----------------------|--|---------------------|-----------------|
| 4* | ** | | | ĺ | |
| 4• | 4 • | | | 1 | |
| | | | | I | |
| | | - | | | |
| | 20 m m n n | 3 | ຉຉຉຉຉຉຉ | ю | <i>с</i> |
| ~ ~ ~ ~ ~ | ~ - ~ ~ - | 2 | 0 0 0 0 0 0 0 0 | 2 | 2 |
| dd ff | dd ff | rr | 6 6 6 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 | rr | rr |
| 38 48 58 68 78 | 37 47 57 67 77 | 24 | 11 15 11 11 11 11 11 | 25 | 27 |
| DIR INH(A) IX1 IX1 IX | DIR INH(A) INH(X) IX1 IX | REL | DIR(b0) DIR(b1) DIR(b2) DIR(b2) DIR(b3) DIR(b3) DIR(b5) DIR(b5) DIR(b7) | REL | REL |
| ▲ ■ ■ 0 C b b 0 | p0 C | ? C=0 | 0 ♦ uW | ? C=1 | ζ Z = 1 |
| Arithmetic Shift Left | Arithmetic Shift Right | Branch if Carry Clear | Clear Bit n in Memory | Branch if Carry Set | Branch if Equal |
| ASL (opr) ASLA ASLX ASLX SSL (opr) | Market (opr) Market (opr) Market (opr) Market (opr) Market (opr) | | CFR u' (obr) CFR u' (obr) On This Product, cescale.com | BCS (rel) | BEQ (rel) |

| | | | re | es | ca | le : | Se | niconductor, I | nq | . |
|---------------------------------|------------|----------------------------|--------------------------|------------------|--------------------------|---------------------------|--------------------------|---|-------------------------|--------------------------|
| de | ပ | | | | | | | | | |
| n Co | Z | | | | | | | * | | |
| Condition Code | z | | | | | | | *• | | |
| Con | - | | | | | | | | | |
| s | I | | | | | | | | | |
| Bytes Cycles | | 3 | 3 | 3 | е | e | ĸ | м 4 0 4 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 8 0 | m | ო |
| Bytes | | 2 | 2 | 2 | 2 | 2 | 2 | 5 - 7 3 3 5 7 | 2 | 2 |
| Machine Coding (hexadecimal) | Operand | rr | л | rr | л | r | L | r # # | rr | ۲۲ |
| Machin (hexad | Opcode | 28 | 29 | 22 | 24 | 2F | 2E | A5 B5 C5 D5 E5 F5 25 25 | 23 | 2C |
| Addressing Mode for | Operand | REL | REL | REL | REL | REL | REL | imim Dir EXT IX2 IX1 IX IX REL | REL | REL |
| Boolean | Expression | j H=0 | 7 H=1 | ? (C + Z) = 0 | , C = 0 | ? IRO Pin=1 | ? <u>IRO</u> Pin=0 | ACCA • M ? C = 1 | ? (C + X) = 1 | 5 1=0 |
| Operation | | Branch if Half Carry Clear | Branch if Half Carry Set | Branch if Higher | Branch if Higher or Same | Branch if IRO Pin is High | Branch if IRO Pin is Low | Bit Test Memory with A Branch if Lower | Branch if Lower or Same | Branch if I Bit is Clear |
| Source | | BHCC (rel) | BHCS (rel) | BHI (rel) | g BHS (rel) | SEIH (rel) | BIL (rel) | (lei) thon On This Produc | BLS (rel) | BMC (rel) |

| r | T | | E | ree | es | ca | le | S | ìe | m | ic | oņ | hdy | ICt | 0 | r., | In | C. | | | |
|-----------------|------------------------|---------------------|----------------|---------------|----------------------------|---------|---------|---------|---------|---------|---------|---------|--------------|--------------------------|---------|---------|---------|---------|---------|---------|---------|
| | | | | | * | , | | | | | | | | (| , | | | | | -e | |
| | | | | | | | | | | | | ~ | | | | | | | | | |
| | | | | | | : | | | | | | | | 1 | | | | | | | |
| 1 | | | | | 1 | | | | | | | | | | | | | | | | |
| | | | 1 | 1 | | | | | | | | | | | | | | | | = | |
| m | с | с | с | с | വ | വ | പ | പ | വ | വ | 2 | 5 | ო | ى ك | പ | പ | വ | വ | 2 | ഹ | 2 |
| 2 | 2 | 2 | 2 | 2 | S | с | ო | ო | ო | ო | ო | 3 | 2 | ო | ო | ო | ო | ო | ო | ო | с |
| | | | | | rr | r | L | r | rr | rr | rr | rr | | rr | r | г | r | r | r | r | rr |
| ۲ | Ľ | ۲ | r | ۲ | pp | pp | pp | pp | pp | pp | pp | pp | r | pp | pp | pp | pp | pp | pp | pp | pp |
| 2B | 2D | 26 | 2A | 20 | 01 | 03 | 05 | 07 | 60 | 0B | Ö | 0F | 21 | 8 | 02 | 04 | 90 | 80 | 0A | 20 | OE |
| REL | REL | REL | REL | REL | DIR(b0) | DIR(b1) | DIR(b2) | DIR(b3) | DIR(b4) | DIR(b5) | DIR(b6) | DIR(b7) | REL | DIR(b0) | DIR(b1) | DIR(b2) | DIR(b3) | DIR(b4) | DIR(b5) | DIR(b6) | DIR(b7) |
| ? N = 1 | 7 = 1 | 5 Z = 0 | 2 N=0 | 7 = 1 | ? Bit n of M = 0 | | | | | | | | ? 1=0 | ? Bit n of M = 1 | | | | | | | |
| Branch if Minus | Branch if I Bit is Set | Branch if Not Equal | Branch if Plus | Branch Always | Branch if Bit n of $M = 0$ | | | | | | | | Branch Never | Branch if Bit n of M = 1 | | | | | | | |
| BMI (rel) | BMS (rel) | BNE (rel) | BPL (rel) | BRA (rel) | RCLR n, (opr) | (rel) | In | for | 'n | ati | on | On | BRN (rel) | BRSET n. (opr) | (rel) | du | ct, | 1 | | | |

| | | F | re | e | sc | a | le | S | er | nie | C-0 | n | dų | ct | or, | In | C. | | | |
|---------------------------------|---------|---------------------|---------|---------|---------|-------------|---------|---------|---------|----------------------|-----|----------------------------|---------------|-----------------------|----------------|-----------|--------|---------|-----------|-----------|
| e | ပ | | | | | | | | | | | | | 0 | | | | | | |
| Cod | Z | | | | | | | | | | | | | | | - | | | | |
| ition | z | | | | | | | | | | | | | | | 0 | | | | |
| Condition Code | - | | | | | | | | | | | | | 1 | 0 | | | : | | |
| | I | | | | | | | | | | | | | 1 | | I | | | | |
| Bytes Cycles | | 2 | പ | പ | വ | വ | പ | ß | 5 | 9 | | | | 2 | 2 | 5 | ო | ო | ග | ß |
| Bytes | | 2 | 2 | 7 | 2 | 2 | 2 | 2 | 2 | 2 | | | | ٢ | 1 | 2 | - | | 2 | - |
| Machine Coding (hexadecimal) | Operand | pp | pp | dd | dd | dd | dd | dd | dd | rr | | | | | | dd | | | Ħ | |
| Machin (hexae | Opcode | 10 | 12 | 14 | 16 | 18 | ٩l | 10 | 1E | ΔA | | | | 98 | 9A | ЗF | 4F | 5F | 6F | 7F |
| Addressing Mode for | Operand | DIR(b0) | DIR(b1) | DIR(b2) | DIR(b3) | DIR(b4) | DIR(b5) | DIR(b6) | DIR(b7) | REL | | | | HNI | HNI | DIR | INH(A) | (X)HNI | IX1 | × |
| Boolean | | Mn ♦ 1 | | | | | | | | PC + PC+0002 | _ | (SP) ♦ PCH; SP ♦ SP – 0001 | PC + PC + Rel | C bit ♦ 0 | bit + 0 | M ♦ 00 | A + 00 | X ♦ 00 | M # 00 | M # 00 |
| Operation | | Set Bit n in Memory | | | | | | | | Branch to Subroutine | | | | Clear C Bit | Clear I Bit | Clear | | | | |
| Source | | BSET n, (opr) | Fo | or I | Мо | re | In | for | ma | States (rel) | n C |)n | Thi | s S S S S | | Str (opr) | CLRA | CLRX | CLR (opr) | CLR (opr) |

| Frees | cale Semi - | conductor, ♥ | Inc. |
|---|---|---|--|
| ₩ | ₩ | ₩ | * |
| · 🗰 | ** | | ↔ |
| | | | |
| | | | |
| 0 6 4 6 4 6 0 | ມດມມ | 0 6 4 6 4 6 7 7 9 7 9 7 9 7 9 7 9 7 9 7 9 7 9 7 9 | വ യ ന ന വ |
| - 7 9 9 7 7 | | - 1 0 0 0 1 | |
| = ت م <u>د</u> ه | σ | = # ₽ 4 • | T |
| ∺ dd th dd tf ee | dd ff | ≓ ee hhd | dd # |
| E E C C 1 E E E E E E E E E E E E E E E | 33 43 53 63 73 | F3 E C C B3 F3 F3 F3 F3 F3 F5 F5 F5 F5 F5 F5 F5 F5 F5 F5 F5 F5 F5 | 3A 5A 6A 7A |
| IMM DIR EXT IX1 IX1 IX1 | DIR INH(A) INH(X) IX1 IX | IMM DIR EXT IX1 IX1 IX1 | DIR INH(A) INH(X) IX1 IX |
| ACCA – M | $M \notin \overline{M} = \$FF - M$ $A \notin \overline{A} = \$FF - A$ $X \notin \overline{X} = \$FF - A$ $M \notin \overline{M} = \$FF - M$ $M \notin \overline{M} = \$FF - M$ | M – X | M ♦ M - 01 A ♦ A - 01 X ♦ X - 01 M ♦ M - 01 M ♦ M - 01 |
| Compare A with Memory | 1's Complement | Compare X with Memory | Decrement DEX (same as DECX) |
| CMP (opr) For M | oprovense Monte Control Monte Control Monte Control Co | (Jdo) Xd On This Produ Treescale.com | DECA DECA DECX DEC (opr) DEC (opr) |

| | | | F | re | e | SC | a | le | S | e | m | ic | Oľ | ١d | u | C | O | r, | In | C | | |
|---------------------------------|------------|----------------------------|-----|-------|-----|-----|----|------------|------------|--------------------|------------|-----------------|-----------|--------|-----|-----|-----|---|----------------------------|--------|---------------------------|---|
| e | ပ | | | | | | | | | | | | | | | | | | | | | |
| Cod | Z | 44 | • | | | | | 44 | • | | | | | | | | | | | | | |
| tion | z | 4 1 | • | | | | | (|) | | | | 1 | - | | | | | | | | |
| Condition Code | - | | | | | · | | | | | | | | , | | | | 1 | | | | |
| ပ | Η | | | | | | | I | | | | | | | | | | | <u>.</u> | ****** | @****** | |
| Bytes Cycles | | 2 | ო | 4 | പ | 4 | 3 | 2 | ო | ო | ٯ | 5 | 2 | ო | 4 | ო | 2 | 2 | 9 | 7 | 9 | പ |
| Bytes | | 2 | 2 | ო | n | 2 | 1 | 2 | - | | 2 | 1 | 2 | ო | e | 2 | 1 | 2 | ო | ო | 2 | - |
| ing al) | and | | | = | Ħ | | | | | | | | | = | Ħ | | | | = | Ħ | - 0 | |
| Machine Coding (hexadecimal) | Operand | := | pp | ЧЧ | ee | ff | | рр | | | Ħ | | pp | ЧЧ | ee | Ħ | | pp | ЧЧ | ee | Ħ | |
| hine | | | | | | | | | | | | | | | | | | | <u> </u> | | | |
| | Opcode | A8 | B8 | 8 | D8 | E8 | F8 | 30 | 4C | 50 | ဥ | 7C | BC | ະ 2 | 2 | EC | FC | BD | 0 0 | | <u> </u> | 6 |
| Addressing Mode for | Operand | NIMI | DIR | EXT | IX2 | IX1 | X | DIR | INH(A) | INH(X) | IX1 | × | DIR | EXT | IX2 | IX1 | IX | DIR | EXT | IX2 | IX1 | × |
| Boolean | Expression | ACCA 🛉 ACCA 🕀 M | | | | | | M ♠ M + 01 | A # A + 01 | X \ X \ 101 | M ♦ M + 01 | $M \neq M + 01$ | PC | | | | | PC $rightarrow$ PC + n (n = 1, 2, or 3) | (SP) ♦ PCL; SP ♦ SP – 0001 | P P | PC Fffective address | |
| Operation | | Exclusive OR A with Memory | | | | | | Increment | | INX (same as INCX) | | | Jump | | | | | Jump to Subroutine | | | | |
| Source Form(c) | | EOR (opr) | | Fo | r I | Mo | re | HENC (opr) | PCA | | Enc (opr | SUC (opr) | OMP (opr) | n T | 'hi | s F | Pro | BSR (opr) | ct, | • | | |

| Source | Operation | Boolean | Addressing Mode for | Machin (hexac | Machine Coding (hexadecimal) | Bytes Cycles | Cycles | с С | ondit | Condition Code | ode |
|-----------|---------------------------|-----------------------|------------------------|------------------|---------------------------------|--------------|----------|------------|-------|----------------|-----------|
| | | Expression | Operand | Opcode | Operand | | | I | - | z | N |
| NEG (opr) | Negates (2's Complement) | M ♠ - M (i.e. 00 - M) | DIR | 30 | dd | 2 | თ | | | • | ₩ ICt. |
| | | A = -A | | 40 | | | ω | | | • | (|
| NEGX | | × ← - × | INH(X) | 50 | | <u> </u> | ω | | | | Duz |
| NEG (opr) | | M ← − M | IX1 | 60 | # | 2 | <u>б</u> | | | | e |
| <u> </u> | | M ♦ - M | × | 70 | | | თ | | | | Th |
| NOP | No Operation | | INH | 9D | | L | 2 | | | | Qn |
| ORA (opr) | Inclusive OR | ACCA ACCA + M | MMI | AA | =: | 2 | 2 | | | | |
| | | | DIR | BA | dd | 2 | ω | -1 | | • | (12* |
| | | | EXT | CA | hh II | ω | 4 | | | | N IS |
| | | | IX2 | DA | ee ff | ω | ന | | | | afa |
| | | | IX1 | ΕA | ff | 2 | 4 | | | | 5-1- |
| | | | IХ | FA | | | ω | | | | 0 |
| ROL (opr) | Rotate Left through Carry | | DIR | 39 | dd | 2 | თ | | | •• | n r M |
| ROLA | | | \sim | 49 | | | ω | | | | (FA |
| ROLX | | | INH(X) | 59 | | | ω | | | | |
| ROL (opr) | | C b7 b0 C | IX1 | 69 | ff | 2 | 6 | | | | |
| ROL (opr | | | × | 79 | | | თ | - - | | | |

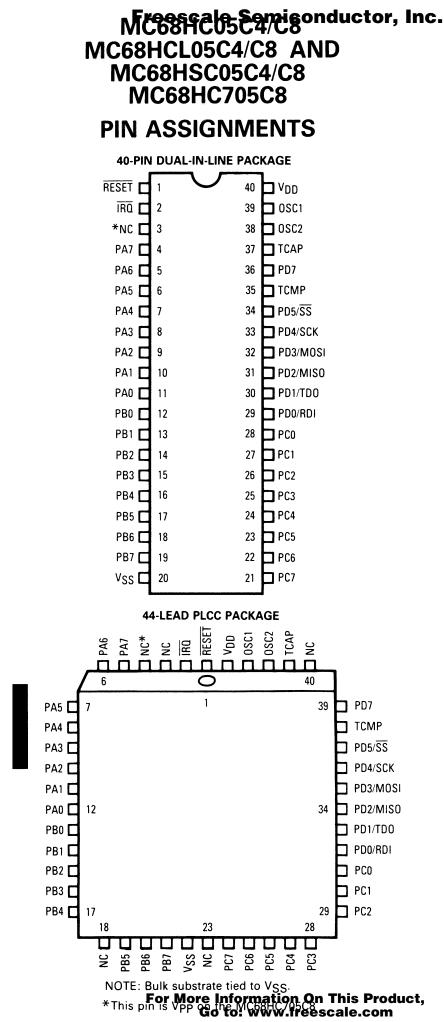
| MUL | LSRA LSRX LSRX LSR (opr) LSR (opr) | I SB (opr) | LSL (opr) | LSL (opr) LSLA LSLX | LDX (opr) | LDA (opr) |
|-------------------|--|---------------------|------------|---------------------------|--|--|
| Unsigned Multiply | | Indical Chift Bight | | Logical Shift Left | Load X from Memory | Load A from Memory |
| X:A | 0 • | | C b7 b0 | | × • M | ACCA 🕇 M |
| ĪŽH | INH(A) | R | × × | DIR INH(A) INH(X) | IMM DIR EXT IX2 IX1 IX1 | IMM EXT IX2 IX |
| 42 | 54 54 74 | 34 | 68 78 | 58 58 | FE E E E E E E | н П П П С С С С С С С С С С С С С С С С |
| | ± 5 | dи | ff | dd | ff = ff = | dd hh ff ff |
| _ | - ~ 1 | 5 | <u> </u> | | | |
| 11 | ၯႍၯၯၯၯ | л | თი | ωωσ | σωαταω | σωμαν |
| 0 | | | | 1 | | |
| | | | | | | |
| | (| > | 1 | 4 • | ++ | ** |
| | ¢1 | • | Mora | 4 | tion On This Bra | 4 |
| 0 | | or | more Go | to: Wy | tion On This Pro w.freescale.con | auci, 1 |

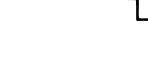
| | es | | e S | emi | cond | lu | ct | or | , | n | C | | |
|---|---------------------|-----------------------|--|--|--|---------------------|----------|----------|----------|-----|----|-----------|-----------|
| ** | | tack) | 4 | | | 41 | | | | | | - | |
| 4* | | m S | 4• | | | 41 | • | | | | | | |
| \$ | | d fro | \$ | | | 41 | • | <u> </u> | | | | | |
| | | (Loaded from Stack) | 4 | | | | | | | | | | 1 |
| | | Ľ | 4• | | | | | | | | | | |
| | 2 | 6 | | | 9 | 2 | <i>с</i> | 4 | 2 | 4 | e | 2 | 2 |
| ~ ~ ~ ~ ~ ~ | - | - | | | - | 2 | 7 | ო | m | 7 | | 1 | Ļ |
| dd ff | | | | | | | q | н Н | ee ff | | | | |
| ± q | | | | | | := | 9 | | ھ | = | | | , , |
| 36 56 66 76 | 9C | 80 | | | 81 | A2 | B2 | C | D2 | E2 | F2 | 66 | 9B |
| DIR INH(A) IXH(X) IX1 IX | HNI | HNI | - | | HNI | MMI | DIR | EXT | IX2 | IX1 | × | INH | HNI |
| C b7 b0 C | SP \$ \$00FF | SP | SP \$SP + 0001; ACCA (SP) SP \$SP + 0001; X \$ (SP) | SP & SP + 0001; PCH (SP) SP & SP + 0001; PCL (SP) | SP ♦ SP + 0001; PCH ♦ (SP) SP ♦ SP + 0001; PCL ♦ (SP) | ACCA A ACCA M C | | | | | | C bit 🛊 1 | I bit ♠ 1 |
| Rotate Right through Carry | Reset Stack Pointer | Return from Interrupt | | | Return from Subroutine | Subtract with Carry | | | | | | Set C Bit | Set I Bit |
| ROR (opr) RORA RORX ROR (opr) ØOR (opr) | MSP | ⊥ r∉ li Go | nfori to: v | | S nºOn 1 freeso | BC (opr) | s P | roe | duc | ət, | | SEC | SEI |

| | | F | r | e | S | ca | le | | m | ic | :0 | nd | | |)r | , | nc | - |
|---------------------------------|------------|-------------------|-----|-----|-----|--------------|-----------------------------|-------------------|----------|-----|-----------|--------|---------------------|----------|-----|-----|-----|---|
| de | С | | | | | | | 0 | <u>.</u> | | | | | • | | | | |
| Coc | Z | 4 | | | | | | () | • | | | | 4 0 | } | | | | |
| ition | z | \$ | | | | | | 4 |) | | | | 41 |) | | | | |
| Condition Code | - | | | | : | | 0 | | · · · · | | | | | | | | | |
| | I | | | | | | | | | | | | 1 | | | | | |
| Bytes Cycles | | 4 | ഹ | 9 | വ | 4 | 2 | 4 | വ | 9 | ß | 4 | 2 | ო | 4 | ß | 4 | ო |
| Bytes | | 2 | с | ო | 2 | , | - | 2 | ო | ო | 2 | 1 | 2 | 2 | ო | ო | 2 | - |
| ng () | and | | Ξ | ff | | | | | = | ff | | | | | = | ff | | |
| Codi cima | Operand | pp | Ļ | e | ÷ | | | pp | ЧЧ | e | ÷ | | | pp | ЧЧ | e | ÷ | |
| Aachine Codin (hexadecimal) | | - | | | | | | | | | | | | _ | | | · | |
| Machine Coding (hexadecimal) | Opcode | Β7 | C7 | D7 | E7 | F7 | 8E | BF | СF | DF | Ш | Ц Ц | AO | BO | 8 | 8 | EO | Ē |
| Addressing Mode for | Operand | DIR | EXT | IX2 | ĬX1 | × | HNI | DIR | EXT | IX2 | XI IXI | × | IMIM | DIR | EXT | IX2 | IX1 | × |
| Boolean | Expression | M + ACCA | | | | | | M∉X | | | | | ACCA A ACCA – M | | | | | |
| Operation | | Store A in Memory | | | | | Enable IRO, Stop Oscillator | Store X in Memory | - | | | | Subtract | | | | | |
| Source | Form(s) | STA (opr) | F | or | Μα | ore | HOL Infe | STX (opr) | ati | ion | 0 | n T | tig UB (opr) | Pr | od | uc | t, | |

| escale Semicon | | Ō | | 1 |
|---|-----------------|--|-----------------|-----------------------------|
| | | ** | | |
| | 1 | ** | 1 | |
| - | | | | 0 |
| | | | | |
| 10 | 2 | 4 m m m 4 | 2 | 2 |
| ~ | | ~ ~ ~ ~ ~ ~ | - | - |
| | | dd ff | | |
| 83 | 97 | 30 50 60 70 | 9F | 8F |
| IZ | INH | DIR INH(A) INH(X) IX1 IX | INH | HNI |
| PC & PC + 0001 (SP) & PCL; SP & SP - 0001 (SP) & X; SP & SP - 0001 (SP) & X; SP & SP - 0001 (SP) & ACCA; SP & SP - 0001 (SP) & CC; SP & SP - 0001 (SP) & CC; SP & SP - 0001 1 bit & 1 PCH & n - 0003 (vector PCL & n - 0002 fetch) | X 🛉 ACCA | 0 - W | ACCA • X | |
| Software Interrupt | Transfer A to X | Test for Negative or Zero | Transfer X to A | Enable Interrupts, Halt CPU |
| ∑ r⊄More Information On T | XAI | LAST (opr) ASTA SSTX TST (opr) TST (opr) | TXA | WAIT |

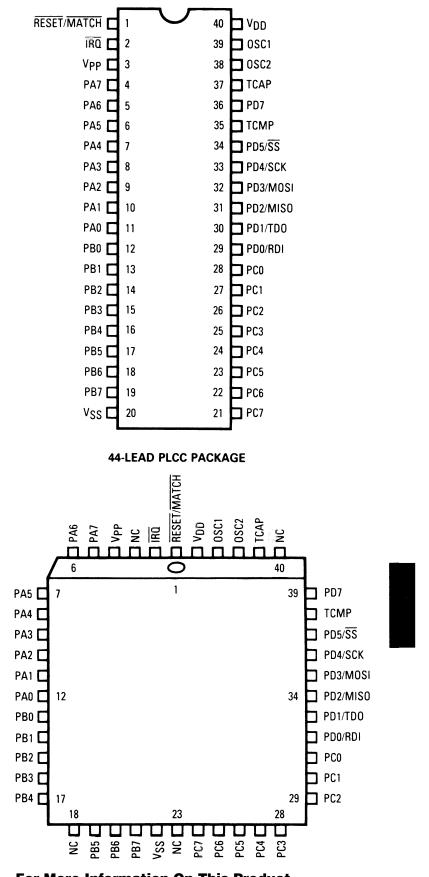
Fr





Freescale Semiconductor, Inc. MC68HC805C4 PIN ASSIGNMENTS

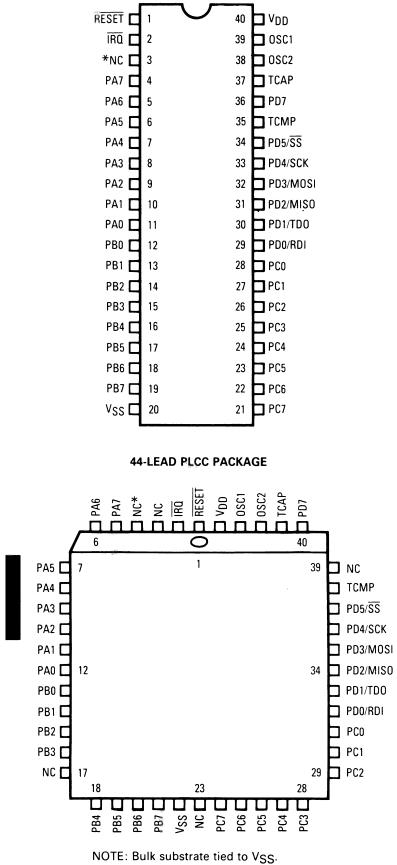
40-PIN DUAL-IN-LINE PACKAGE



For More Information On This Product, Go to: www.freescale.com

Ereescale Semiconductor, Inc. MC68HC05C9 (ONLY) PIN ASSIGNMENTS

40-PIN DUAL-IN-LINE PACKAGE



*This pin is VPP Go to: WWW.freescale.com

Freescale Semiconductor, Inc. ASCII CHART

| | 7 | ФГ о + J > З X > и { ј | DEL |
|------------------|--------------------------|---|-----|
| | 9 | - פסטםפרטד | 0 |
| | 5 | LOKSHコ>w××ν-/-< | 1 |
| SET (7-Bit Code) | 4 | ϣϥϐϢϤ϶ϤϿ϶϶Ͽ | 0 |
| | m | Ο-οσφονν ΙΙΛ | ~: |
| ASCII CHARACTER | 7 | Ŋ→: ≉⇔%&、~~★+~I ・ | / |
| ASCII CH | 1 | DLE DC1 DC2 DC3 DC4 SVN SVN SVN SVN SVN SVN SVN SVN SVN SVN | NS |
| | 0 | NUL SOH STX ACK BEL BBEL CR CR CR CR SO | SI |
| | MS LS Dig. Dig. | о-см4らら と ⊗の∢ВООШ | LL. |

Freescale Semiconductor, Inc. HEX/DEC CONVERSION

HEXADECIMAL AND DECIMAL CONVERSION

How to use:

Conversion to Decimal: Find the decimal weights for corresponding hexadecimal characters beginning with the least significant character. The sum of the decimal weights is the decimal value of the hexadecimal number.

Conversion to Hexadecimal: Find the highest decimal value in the table which is lower than or equal to the decimal number to be converted. The corresponding hexadecimal character is the most significant. Subtract the decimal value found from the decimal number to be converted. With the difference repeat the process to find subsequent hexadecimal characters.

| 15 | Byte | te | 8 | 7 | | Ву | Byte | | 0 |
|-----------------|---------|-----|-------|-----|------|----------------|------|------|-----|
| 15 | Char 12 | 11 | œ | 7 | Char | 4 | 3 | Char | 0 |
| XaHe | Dec | Hex | Dec | Нех | | Dec | Нех | | Dec |
| 0 r M | 0 | 0 | 0 | 0 | | 0 | 0 | | 0 |
| ore | 4,096 | - | 256 | - | | 16 | - | | - |
| 5 11 e 20 | 8,192 | 2 | 512 | 2 | | 32 | 2 | | 2 |
| | 12,228 | ю | 768 | ო | | 8 4 | ო | | ო |
| 4 rm | 16,384 | 4 | 1,024 | 4 | | 2 | 4 | | 4 |
| | 20,480 | £ | 1,280 | 5 | | 8 | 5 | | 2 |
| ت ion | 24,576 | 6 | 1,536 | 9 | | 8 | 9 | | 9 |
| 0 | 28,672 | 7 | 1,792 | 7 | | 112 | 7 | | 7 |
| | 32,768 | 8 | 2,048 | ω | | 128 | 8 | | 8 |
| | 36,864 | 6 | 2,304 | 6 | | 1 4 | 6 | | 6 |
| ⊲ sF | 40,960 | A | 2,560 | ∢ | | 160 | ۷ | | 10 |
| ے Pro | 45,056 | В | 2,816 | в | | 176 | В | | 1 |
| ാ du | 49,152 | C | 3,072 | ပ | | 192 | ပ | | 12 |
| ∩ ct, | 53,248 | D | 3,328 | ۵ | | 208 | ۵ | | 13 |
| ш | 57,344 | ш | 3,584 | ш | | 224 | ш | | 14 |
| ш | 61,440 | ш | 3,840 | u. | | 240 | ш | | 15 |

Downloaded from Elcodis.com electronic components distributor

For More Information On This Product, Go to www.fireescale.com

For More Informatipg on This Product, Go to: www.ireescale.com



MECHANICAL DATA

HEX/DEC CONVERSION ASCII CHART

For More Information On This Product, Go to: www.freescale.com



MEMORY MAPS

REGISTER/CONTROL BIT ASSIGNMENTS



INSTRUCTIONS ADDRESSING MODES EXECUTION TIMES

MECHANICAL DATA

HEX/DEC CONVERSION ASCII CHART The MC68HC05 Family of HCMOS devices covered in this reference guide are as follows:

MC68HC05C4 MC68HC05C8 MC68HC05C9 MC68HC705C8 MC68HC805C4 MC68HCL05C4 MC68HCL05C8 MC68HSC05C4

Motorola reserves the right to make changes without further notice to any product herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in Motorola data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Motorola does not convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola products could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part. Motorola and 😬 are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

Mfax is a trademark of Motorola, Inc.

How to reach us:

USA/EUROPE/Locations Not Listed: Motorola Literature Distribution: P.O. Box 5405, Denver, Colorado 80217. 303-675-2140 or 1-800-441-2447 Mfax[™]. RMFAX0@email.sps.mot.com - TOUCHTONE 602-244-6609 - US & Canada ONLY 1-800-774-1848 INTERNET: http://motorola.com/sps

JAPAN: Nippon Motorola Ltd.; SPD, Strategic Planning Office, 4-32-1, Nishi* Gotanda, Shinagawa-ku, Tokyo 141, Japan. 81-3-5487-8488

ASIA/PACIFIC: Motorola Semiconductors H.K. Ltd.; 8B Tai Ping Industrial Park, 51 Ting Kok Road, Tai Po, N.T., Hong Kong. 852-26629298



For More Information On This Product,

1ATX23834-3 PRINTED IN USA 5/9' IMPERIAL LITRO 29621 3,000 LITCSIC UCCL

Downloaded from Elcodis.com electronic components distributor