Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

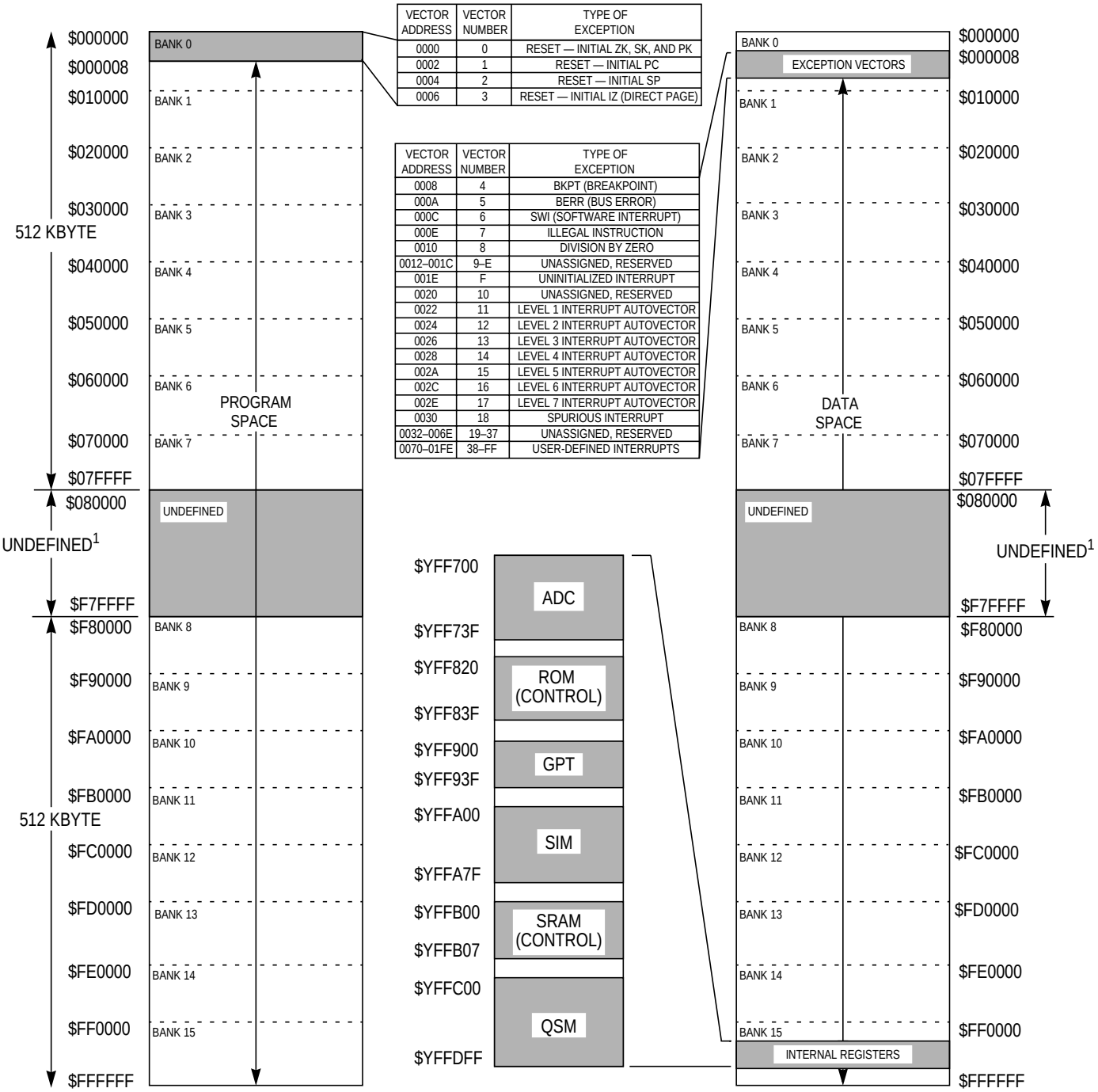
Details	
Product Status	Active
Core Processor	CPU16
Core Size	16-Bit
Speed	16MHz
Connectivity	EBI/EMI, SCI, SPI
Peripherals	POR, PWM, WDT
Number of I/O	16
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc68ck16z1cag16

TABLE OF CONTENTS
(Continued)

Paragraph Title Page

SECTION 11
GENERAL-PURPOSE TIMER

LIST OF TABLES
(Continued)



NOTE:
1. THE ADDRESSES DISPLAYED IN THIS MEMORY MAP ARE THE FULL 24-BIT IMB ADDRESSES. THE CPU16 ADDRESS BUS IS 20 BITS WIDE, AND CPU16 ADDRESS LINE 19 DRIVES IMB ADDRESS LINES [23:20]. THE BLOCK OF ADDRESSES FROM $080000 TO $F7FFFF MARKED AS UNDEFINED WILL NEVER APPEAR ON THE IMB. MEMORY BANKS 0 TO 15 APPEAR FULLY CONTIGUOUS IN THE CPU16'S FLAT 20-BIT ADDRESS SPACE. THE CPU16 NEED ONLY GENERATE A 20-BIT EFFECTIVE ADDRESS TO ACCESS ANY LOCATION IN THIS RANGE.

Figure 3-15 MC68HC16Z2/Z3 Separate Program and Data Space Map

SECTION 4
CENTRAL PROCESSOR UNIT

This section is an overview of the central processor unit (CPU16). For detailed information, refer to the *CPU16 Reference Manual* (CPU16RM/AD).

4.1 General

The CPU16 provides compatibility with the M68HC11 CPU and also provides additional capabilities associated with 16- and 32-bit data sizes, 20-bit addressing, and digital signal processing. CPU16 registers are an integral part of the CPU and are not addressed as memory locations.

The CPU16 treats all peripheral, I/O, and memory locations as parts of a linear one Megabyte address space. There are no special instructions for I/O that are separate from instructions for addressing memory. Address space is made up of sixteen 64-Kbyte banks. Specialized bank addressing techniques and support registers provide transparent access across bank boundaries.

The CPU16 interacts with external devices and with other modules within the microcontroller via a standardized bus and bus interface. There are bus protocols used for memory and peripheral accesses, as well as for managing a hierarchy of interrupt priorities.

4.2 Register Model

Figure 4-1 shows the CPU16 register model. Refer to the paragraphs that follow for a detailed description of each register.

Table 4-5 Exception Vector Table

Vector Number	Vector Address	Address Space	Type of Exception
0	0000	P	Reset — Initial ZK, SK, and PK
	0002	P	Reset — Initial PC
	0004	P	Reset — Initial SP
	0006	P	Reset — Initial IZ (Direct Page)
4	0008	D	Breakpoint
5	000A	D	Bus Error
6	000C	D	Software Interrupt
7	000E	D	Illegal Instruction
8	0010	D	Division by Zero
9 – E	0012 – 001C	D	Unassigned, Reserved
F	001E	D	Uninitialized Interrupt
10	0020	D	Unassigned, Reserved
11	0022	D	Level 1 Interrupt Autovector
12	0024	D	Level 2 Interrupt Autovector
13	0026	D	Level 3 Interrupt Autovector
14	0028	D	Level 4 Interrupt Autovector
15	002A	D	Level 5 Interrupt Autovector
16	002C	D	Level 6 Interrupt Autovector
17	002E	D	Level 7 Interrupt Autovector
18	0030	D	Spurious Interrupt
19 – 37	0032 – 006E	D	Unassigned, Reserved
38 – FF	0070 – 01FE	D	User-Defined Interrupts

4.13.2 Exception Stack Frame

During exception processing, the contents of the program counter and condition code register are stacked at a location pointed to by SK : SP. Unless it is altered during exception processing, the stacked PK : PC value is the address of the next instruction in the current instruction stream, plus $0006. **Figure 4-6** shows the exception stack frame.

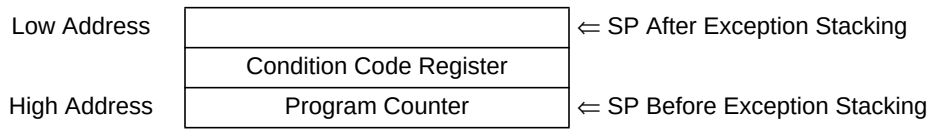


Figure 4-6 Exception Stack Frame Format

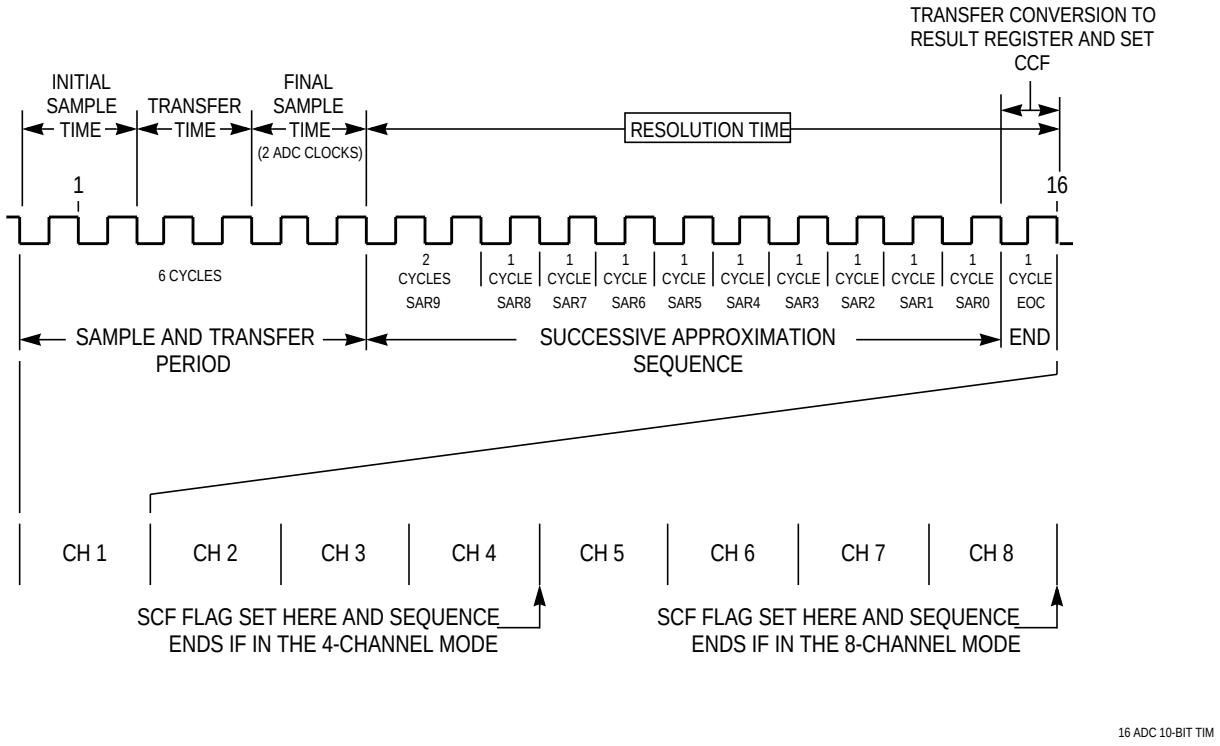


Figure 8-3 10-Bit Conversion Timing

8.7.7 Successive Approximation Register

The successive approximation register (SAR) accumulates the result of each conversion one bit at a time, starting with the most significant bit.

At the start of the resolution period, the MSB of the SAR is set, and all less significant bits are cleared. Depending on the result of the first comparison, the MSB is either left set or cleared. Each successive bit is set or left cleared in descending order until all eight or ten bits have been resolved.

When conversion is complete, the content of the SAR is transferred to the appropriate result register. Refer to **APPENDIX D REGISTER SUMMARY** for register mapping and configuration.

8.7.8 Result Registers

Result registers are used to store data after conversion is complete. The registers can be accessed from the IMB under ABIU control. Each register can be read from three different addresses in the ADC memory map. The format of the result data depends on the address from which it is read. **Table 8-9** shows the three types of formats.

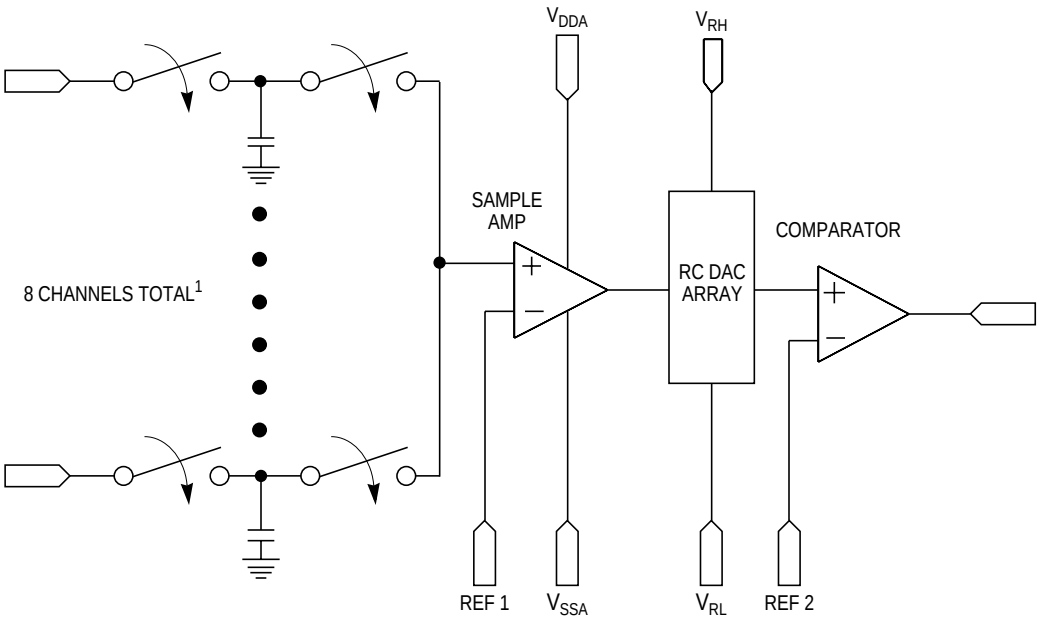


Figure 8-4 Analog Input Circuitry

Since the sample amplifier is powered by V_{DDA} and V_{SSA}, it can accurately transfer input signal levels up to but not exceeding V_{DDA} and down to but not below V_{SSA}. If the input signal is outside of this range, the output from the sample amplifier is clipped.

In addition, V_{RH} and V_{RL} must be within the range defined by V_{DDA} and V_{SSA}. As long as V_{RH} is less than or equal to V_{DDA}, and V_{RL} is greater than or equal to V_{SSA}, and the sample amplifier has accurately transferred the input signal, resolution is ratiometric within the limits defined by V_{RL} and V_{RH}. If V_{RH} is greater than V_{DDA}, the sample amplifier can never transfer a full-scale value. If V_{RL} is less than V_{SSA}, the sample amplifier can never transfer a zero value.

Figure 8-5 shows the results of reference voltages outside the range defined by V_{DDA} and V_{SSA}. At the top of the input signal range, V_{DDA} is 10 mV lower than V_{RH}. This results in a maximum obtainable 10-bit conversion value of 3FE. At the bottom of the signal range, V_{SSA} is 15 mV higher than V_{RL}, resulting in a minimum obtainable 10-bit conversion value of three.

Grounding is the most important factor influencing analog circuit performance in mixed signal systems (or in stand-alone analog systems). Close attention must be paid to avoid introducing additional sources of noise into the analog circuitry. Common sources of noise include ground loops, inductive coupling, and combining digital and analog grounds together inappropriately.

The problem of how and when to combine digital and analog grounds arises from the large transients which the digital ground must handle. If the digital ground is not able to handle the large transients, the current from the large transients can return to ground through the analog ground. It is the excess current overflowing into the analog ground which causes performance degradation by developing a differential voltage between the true analog ground and the microcontroller's ground pin. The end result is that the ground observed by the analog circuit is no longer true ground and often ends in skewed results.

Two similar approaches designed to improve or eliminate the problems associated with grounding excess transient currents involve star-point ground systems. One approach is to star-point the different grounds at the power supply origin, thus keeping the ground isolated. Refer to **Figure 8-6**.

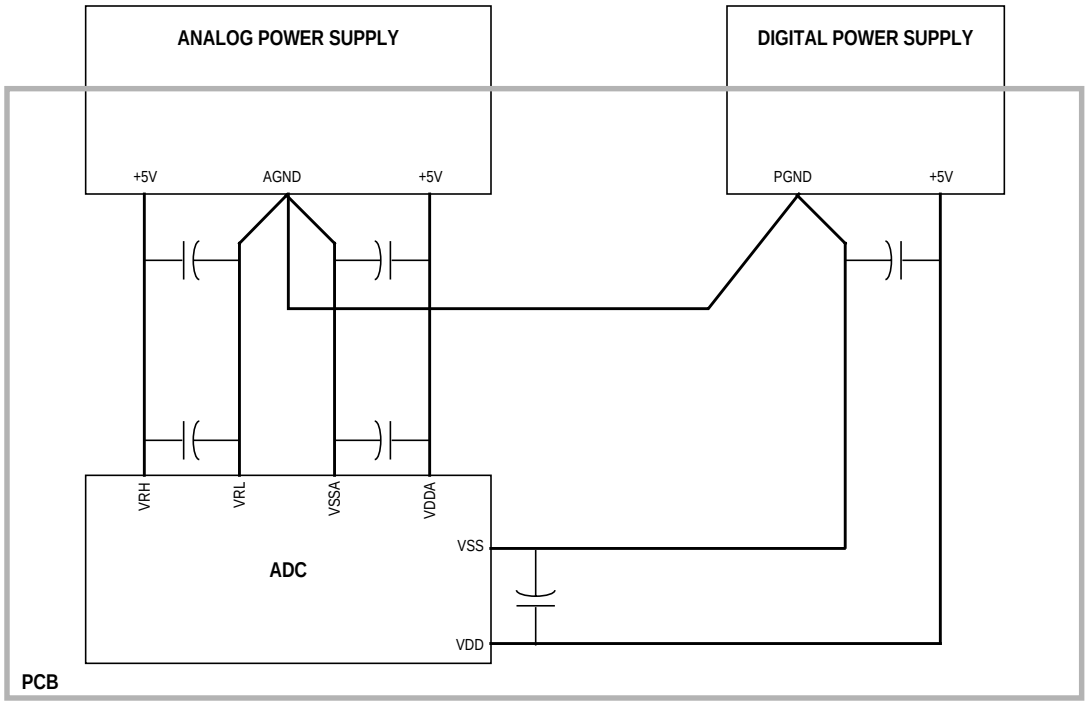


Figure 8-6 Star-Ground at the Point of Power Supply Origin

Another approach is to star-point the different grounds near the analog ground pin on the microcontroller by using small traces for connecting the non-analog grounds to the analog ground. The small traces are meant only to accommodate DC differences, not AC transients.

The current out of the pin (I_{OUT}) under negative stress is determined by the following equation:

$$I_{OUT} = \frac{|V_{STRESS} - V_{BE}|}{R_{STRESS}}$$

where:

V_{STRESS} = Adjustable voltage source

V_{BE} = Parasitic bipolar base/emitter voltage (refer to $V_{NEGCLAMP}$ in **APPENDIX A ELECTRICAL CHARACTERISTICS**)

R_{STRESS} = Source impedance (10K resistor in **Figure 8-7** on stressed channel)

The current into (I_{IN}) the neighboring pin is determined by the $1/K_N$ (Gain) of the parasitic bipolar transistor ($1/K_N << 1$).

One way to minimize the impact of stress conditions on the ADC is to apply voltage limiting circuits such as diodes to supply and ground. However, leakage from such circuits and the potential influence on the sampled voltage to be converted must be considered. Refer to **Figure 8-8**.

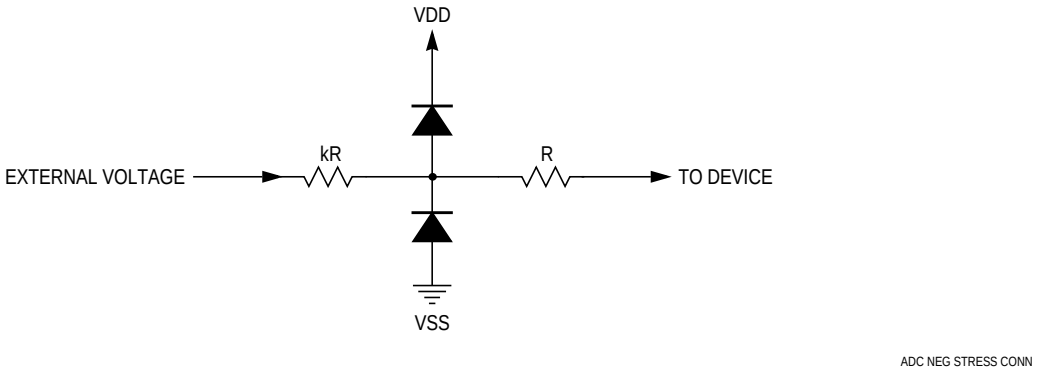


Figure 8-8 Voltage Limiting Diodes in a Negative Stress Circuit

Another method for minimizing the impact of stress conditions on the ADC is to strategically allocate ADC inputs so that the lower accuracy inputs are adjacent to the inputs most likely to see stress conditions.

Finally, suitable source impedances should be selected to meet design goals and minimize the effect of stress conditions.

8.8.5 Analog Input Considerations

The source impedance of the analog signal to be measured and any intermediate filtering should be considered whether external multiplexing is used or not. **Figure 8-9** shows the connection of eight typical analog signal sources to one ADC analog input pin through a separate multiplexer chip. Also, an example of an analog signal source connected directly to a ADC analog input channel is displayed.

Data transfer is synchronized with the internally-generated serial clock SCK. Control bits, CPHA and CPOL, in SPCR0, control clock phase and polarity. Combinations of CPHA and CPOL determine upon which SCK edge to drive outgoing data from the MOSI pin and to latch incoming data from the MISO pin.

Baud rate is selected by writing a value from two to 255 into SPBR[7:0] in SPCR0. The QSPI uses a modulus counter to derive the SCK baud rate from the MCU system clock.

The following expressions apply to the SCK baud rate:

$$\text{SCK Baud Rate} = \frac{f_{sys}}{2 \times \text{SPBR[7:0]}}$$

or

$$\text{SPBR[7:0]} = \frac{f_{sys}}{2 \times \text{SCK Baud Rate Desired}}$$

Giving SPBR[7:0] a value of zero or one disables the baud rate generator and SCK assumes its inactive state.

The DSCK bit in each command RAM byte inserts either a standard (DSCK = 0) or user-specified (DSCK = 1) delay from chip-select assertion until the leading edge of the serial clock. The DSCKL field in SPCR1 determines the length of the user-defined delay before the assertion of SCK. The following expression determines the actual delay before SCK:

$$\text{PCS to SCK Delay} = \frac{\text{DSCKL[6:0]}}{f_{sys}}$$

where DSCKL[6:0] equals {1, 2, 3,..., 127}.

When DSCK equals zero, DSCKL[6:0] is not used. Instead, the PCS valid-to-SCK transition is one-half the SCK period.

There are two transfer length options. The user can choose a default value of eight bits, or a programmed value from eight to sixteen bits, inclusive. The programmed value must be written into BITS[3:0] in SPCR0. The BITSE bit in each command RAM byte determines whether the default value (BITSE = 0) or the BITS[3:0] value (BITSE = 1) is used. **Table 9-3** shows BITS[3:0] encoding.

9.4.2 SCI Pins

Two unidirectional pins, TXD (transmit data) and RXD (receive data), are associated with the SCI. TXD can be used by the SCI or for general-purpose I/O. TXD function is controlled by PQSPA7 in the port QS pin assignment register (PQSPAR) and TE in SCI control register 1 (SCCR1). The receive data (RXD) pin is dedicated to the SCI.

9.4.3 SCI Operation

The SCI can operate in polled or interrupt-driven mode. Status flags in SCSR reflect SCI conditions regardless of the operating mode chosen. The TIE, TCIE, RIE, and ILIE bits in SCCR1 enable interrupts for the conditions indicated by the TDRE, TC, RDRF, and IDLE bits in SCSR, respectively.

9.4.3.1 Definition of Terms

- Bit-Time — The time required to transmit or receive one bit of data, which is equal to one cycle of the baud frequency.
- Start Bit — One bit-time of logic zero that indicates the beginning of a data frame. A start bit must begin with a one-to-zero transition and be preceded by at least three receive time samples of logic one.
- Stop Bit — One bit-time of logic one that indicates the end of a data frame.
- Frame — A complete unit of serial information. The SCI can use 10-bit or 11-bit frames.
- Data Frame — A start bit, a specified number of data or information bits, and at least one stop bit.
- Idle Frame — A frame that consists of consecutive ones. An idle frame has no start bit.
- Break Frame — A frame that consists of consecutive zeros. A break frame has no stop bits.

9.4.3.2 Serial Formats

All data frames must have a start bit and at least one stop bit. Receiving and transmitting devices must use the same data frame format. The SCI provides hardware support for both 10-bit and 11-bit frames. The M bit in SCCR1 specifies the number of bits per frame.

The most common data frame format for NRZ serial interfaces is one start bit, eight data bits (LSB first), and one stop bit; a total of ten bits. The most common 11-bit data frame contains one start bit, eight data bits, a parity or control bit, and one stop bit. Ten-bit and 11-bit frames are shown in **Table 9-4**.

10.3.4.2 CPHA = 1 Transfer Format

Figure 10-4 is a timing diagram of an 8-bit, MSB-first SPI transfer in which CPHA equals one. Two waveforms are shown for SCK, one for CPOL equal to zero and another for CPOL equal to one. The diagram may be interpreted as a master or slave timing diagram since the SCK, MISO and MOSI pins are directly connected between the master and the slave. The MISO signal shown is the output from the slave and the MOSI signal shown is the output from the master. The $\overline{SS}$ line is the slave select input to the slave.

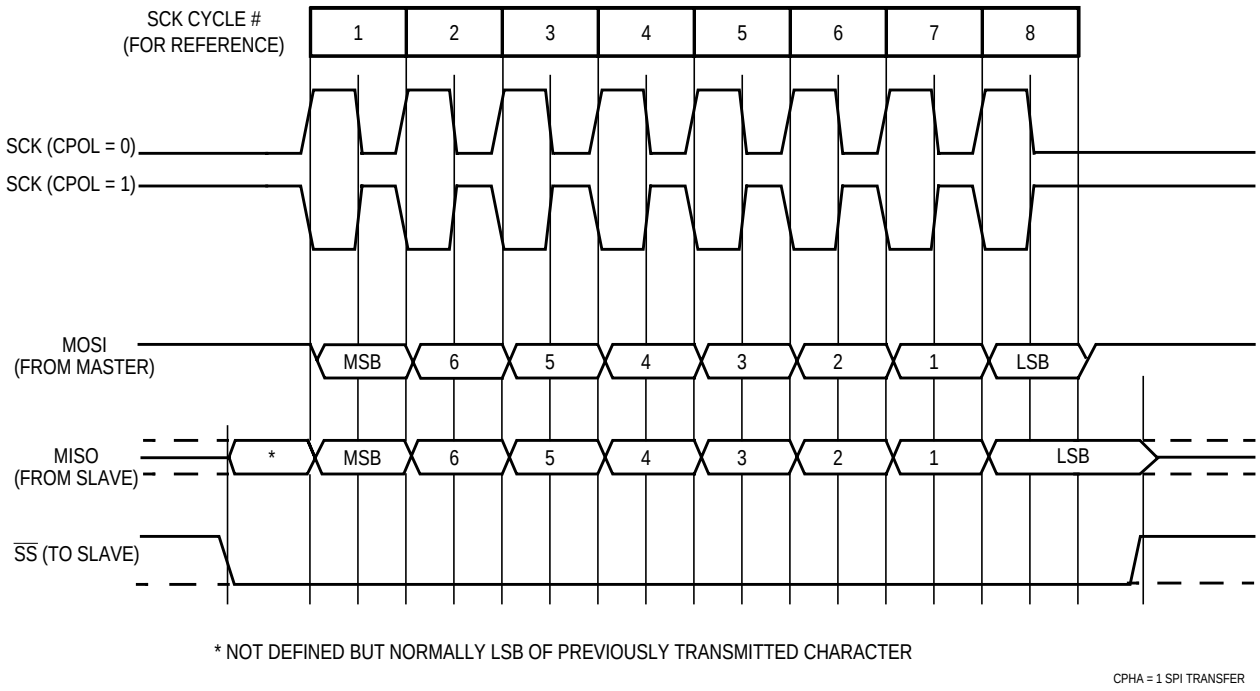


Figure 10-4 CPHA = 1 SPI Transfer Format

For a master, writing to the SPDR initiates the transfer. For a slave, the first edge of SCK indicates the start of a transfer. The SPI is left-shifted on the first and each succeeding odd clock edge, and data is latched on the second and succeeding even clock edges.

SCK is inactive for the last half of the eighth SCK cycle. For a master, SPIF is set at the end of the eighth SCK cycle (after the seventeenth SCK edge). Since the last SCK edge occurs in the middle of the eighth SCK cycle, however, the slave has no way of knowing when the end of the last SCK cycle occurs. The slave therefore considers the transfer complete after the last bit of serial data has been sampled, which corresponds to the middle of the eighth SCK cycle.

When CPHA is one, the $\overline{SS}$ line may remain at its active low level between transfers. This format is sometimes preferred in systems having a single fixed master and only one slave that needs to drive the MISO data line.

SECTION 11
GENERAL-PURPOSE TIMER

This section is an overview of the general-purpose timer (GPT) function. Refer to the *GPT Reference Manual* (GPTRM/AD) for complete information about the GPT module.

11.1 General

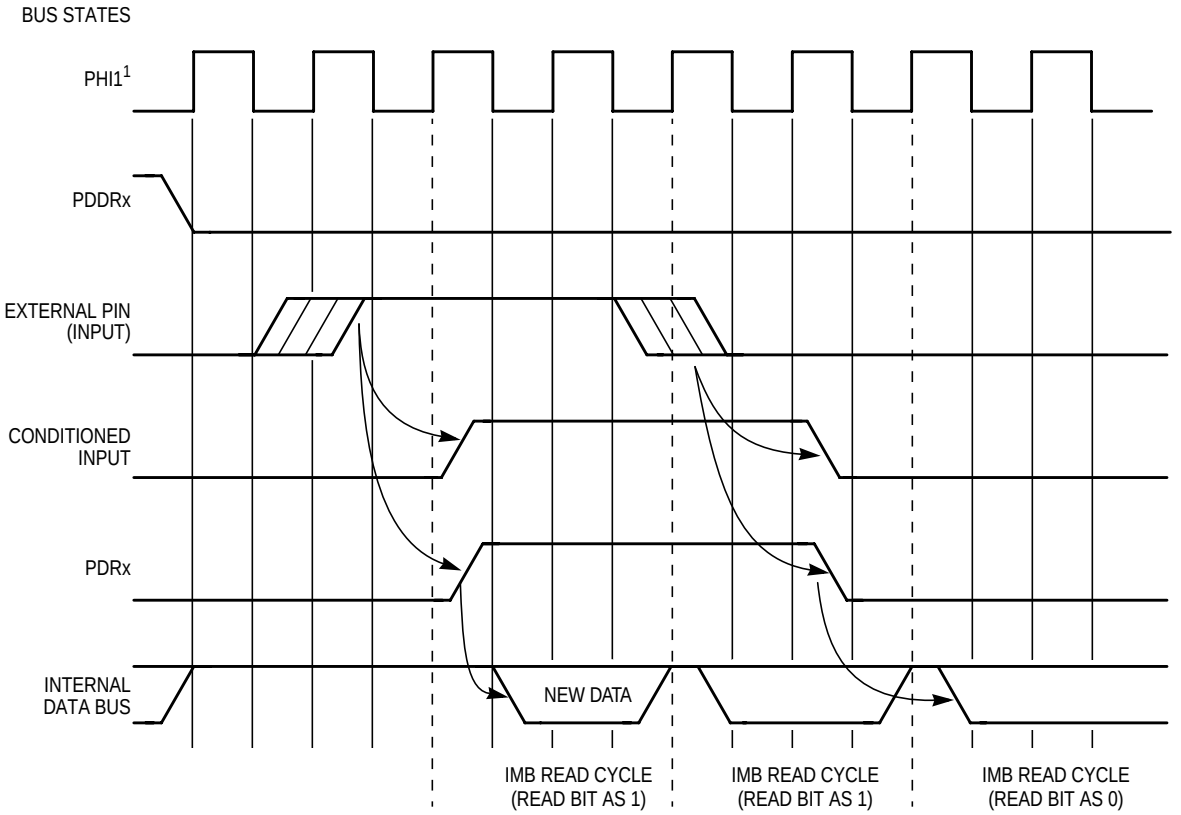
The 11-channel general-purpose timer (GPT) is used in systems where a moderate level of CPU control is required. The GPT consists of a capture/compare unit, a pulse accumulator, and two pulse-width modulators. A bus interface unit connects the GPT to the intermodule bus (IMB). **Figure 11-1** is a block diagram of the GPT.

The capture/compare unit features three input capture channels, four output compare channels, and one channel that can be selected as input capture or output compare. These channels share a 16-bit free-running counter (TCNT) that derives its clock from a nine-stage prescaler or from the external clock input signal, PCLK.

Pulse accumulator channel logic includes an 8-bit counter. The pulse accumulator can operate in either event counting mode or gated time accumulation mode.

Pulse-width modulator outputs are periodic waveforms whose duty cycles can be independently selected and modified by user software. The PWM circuits share a 16-bit free-running counter that can be clocked by the same nine-stage prescaler used by the capture/compare unit or by the PCLK input.

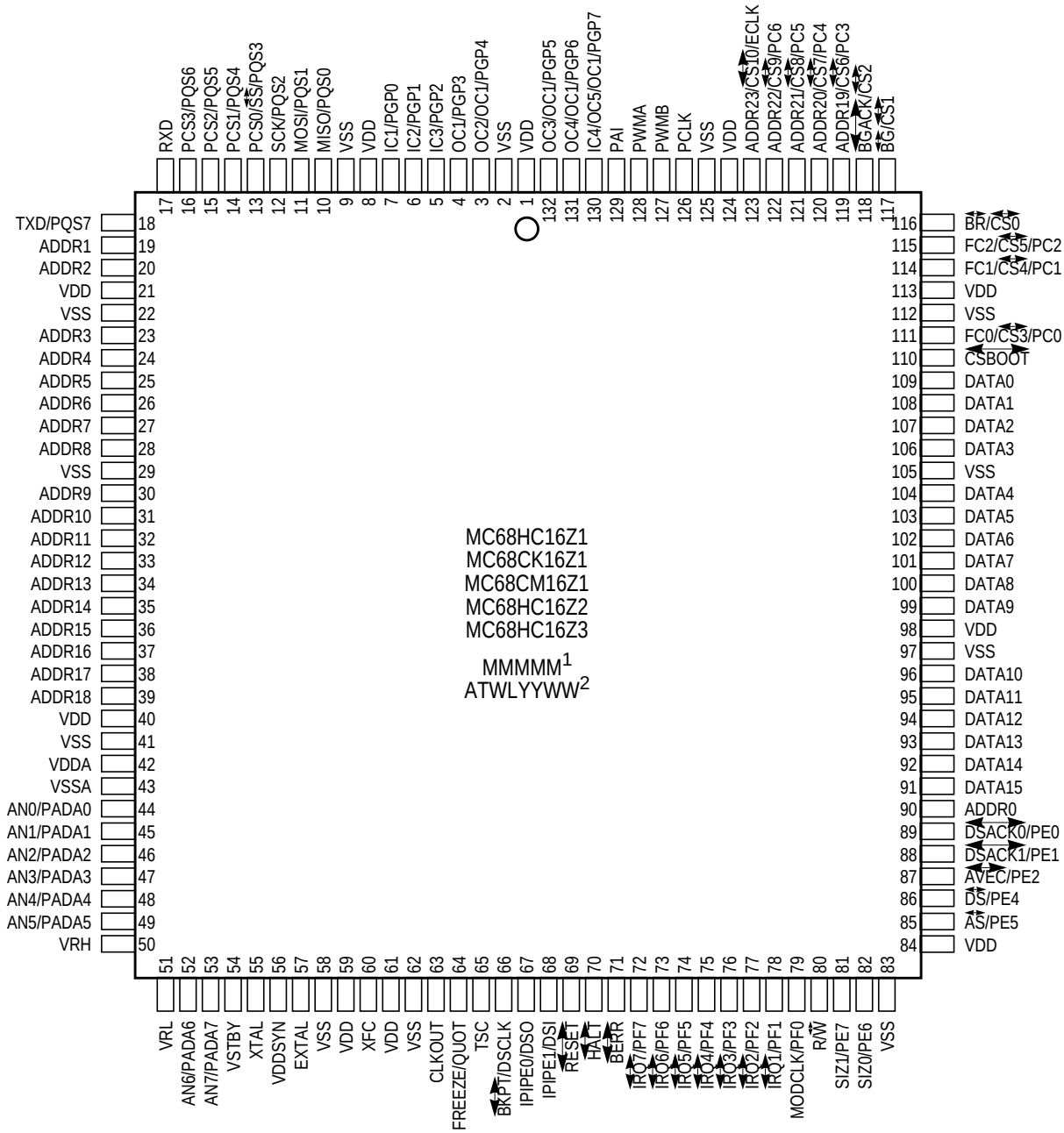
All GPT pins can also be used for general-purpose input/output. The input capture and output compare pins form a bidirectional 8-bit parallel port (port GP). PWM pins are outputs only. PAI and PCLK pins are inputs only.



NOTES:
1. PHI1 IS THE SAME FREQUENCY AS THE SYSTEM CLOCK; HOWEVER, IT DOES NOT HAVE THE SAME TIMING.

GENERAL PURPOSE INPUT

Figure A-31 General-Purpose Input



NOTES:
1. MMMMM = MASK OPTION NUMBER
2. ATWLYYWW = ASSEMBLY TEST LOCATION/YEAR, WEEK

HC16Z1/CKZ1/CMZ1/Z2/Z3 132-PIN QFP

Figure B-1 MC68HC16Z1/CKZ1/CMZ1/Z2/Z3 Pin Assignments for 132-Pin Package

C.2 M68MEVB1632 Modular Evaluation Board

The M68MEVB1632 Modular Evaluation Board (MEVB) is a development tool for evaluating M68HC16 and M68300 MCU-based systems. The MEVB consists of the M68MPFB1632 modular platform board, an MCU personality board (MPB), an in-circuit debugger (ICD16 or ICD32), and development software. MEVB features include:

- An economical means of evaluating target systems incorporating M68HC16 and M68300 HCMOS MCU devices
- Expansion memory sockets for installing RAM, EPROM, or EEPROM
  - — Data RAM: 32K x 16, 128K x 16, or 512K x 16
  - — EPROM/EEPROM: 32K x 16, 64K x 16, 128K x 16, 256K x 16, or 512K x 16
  - — Fast RAM: 32K x 16 or 128K x 16
- Background-mode operation, for detailed operation from a personal computer platform without an on-board monitor
- Integrated assembly/editing/evaluation/programming environment for easy development
- As many as seven software breakpoints
- Re-usable ICD hardware for your target application debug or control
- Two RS-232C terminal input/output (I/O) ports for user evaluation of the serial communication interface
- Logic analyzer pod connectors
- Port replacement unit (PRU) to rebuild I/O ports lost to address/data/control
- On-board V_{PP} (+12 VDC) generation for MCU and flash EEPROM programming.
- On-board wire-wrap area

NOTE

The MC68HC16Z1 and the MC68HC16Z2/Z3 both utilize the M68HC16MPFB, however, each MCU uses a different personality board (M68MPB16Z1 on the MC68HC16Z1; M68MPB16Z2/Z3 on the MC68HC16Z2/Z3).

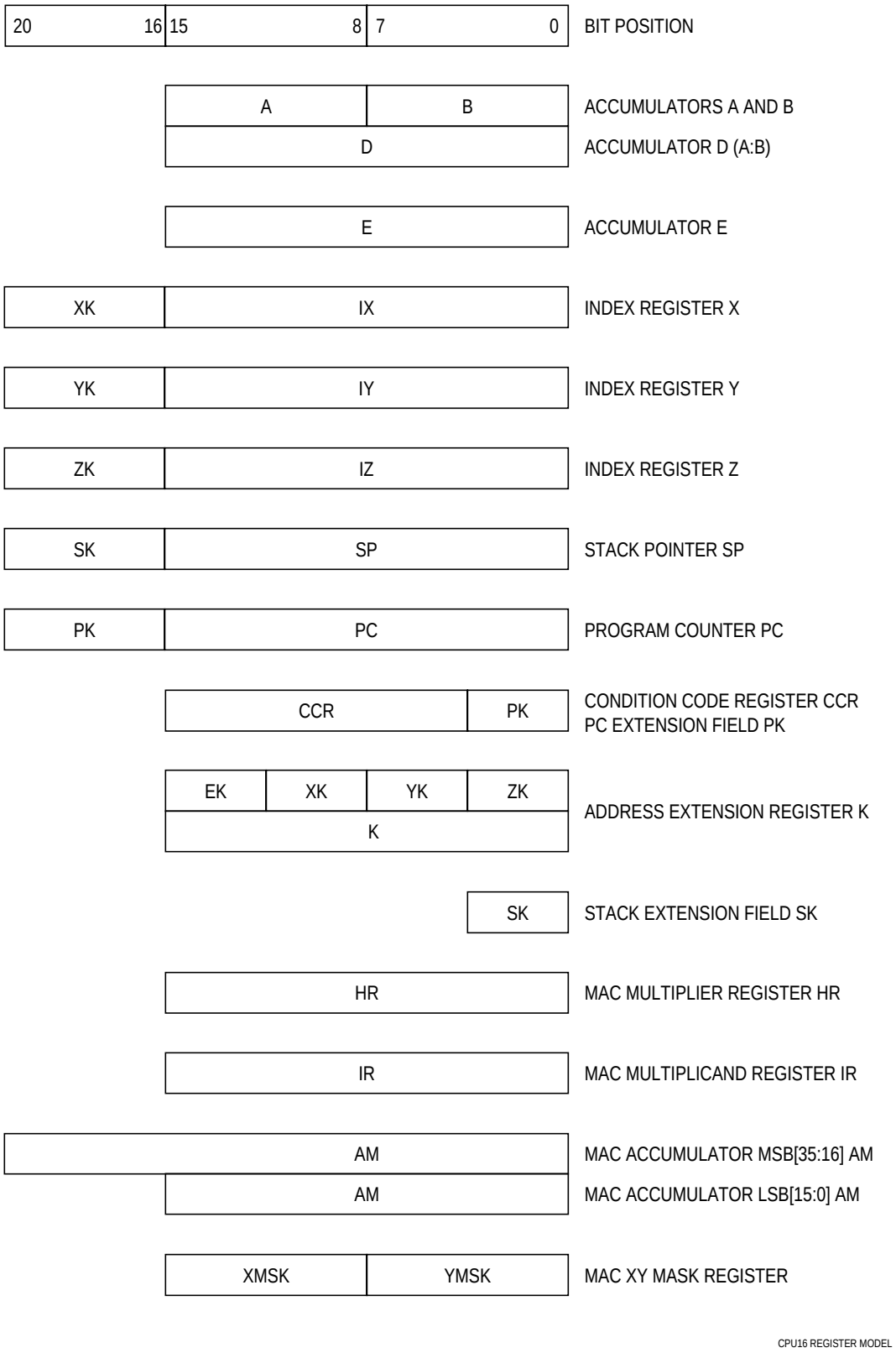


Figure D-1 CPU16 Register Model

D.2 System Integration Module

Table D-2 shows the SIM address map.

Table D-2 SIM Address Map

Address[1]	15 8	7 0
$YFFA00	SIM Module Configuration Register (SIMCR)	
$YFFA02	SIM Test Register (SIMTR)	
$YFFA04	Clock Synthesizer Control Register (SYNCR)	
$YFFA06	Not Used	Reset Status Register (RSR)
$YFFA08	SIM Test Register E (SIMTRE)	
$YFFA0A	Not Used	Not Used
$YFFA0C	Not Used	Not Used
$YFFA0E	Not Used	Not Used
$YFFA10	Not Used	Port E Data Register 0 (PORTE0)
$YFFA12	Not Used	Port E Data Register 1(PORTE1)
$YFFA14	Not Used	Port E Data Direction Register (DDRE)
$YFFA16	Not Used	Port E Pin Assignment Register (PEPAR)
$YFFA18	Not Used	Port F Data Register 0 (PORTF0)
$YFFA1A	Not Used	Port F Data Register 1 (PORTF1)
$YFFA1C	Not Used	Port F Data Direction Register (DDRF)
$YFFA1E	Not Used	Port F Pin Assignment Register (PFPAR)
$YFFA20	Not Used	System Protection Control Register (SYPCR)
$YFFA22	Periodic Interrupt Control Register (PICR)	
$YFFA24	Periodic Interrupt Timer Register (PITR)	
$YFFA26	Not Used	Software Watchdog Service Register (SWSR)
$YFFA28	Not Used	
$YFFA2A	Not Used	
$YFFA2C	Not Used	
$YFFA2E	Not Used	
$YFFA30	Test Module Master Shift A Register (TSTMSRA)	
$YFFA32	Test Module Master Shift B Register (TSTMSRB)	
$YFFA34	Test Module Shift Count Register (TSTSC)	
$YFFA36	Test Module Repetition Counter Register (TSTRC)	
$YFFA38	Test Module Control Register (CREG)	
$YFFA3A	Test Module Distributed Register (DREG)	
$YFFA3C	Not Used	
$YFFA3E	Not Used	
$YFFA40	Not Used	Port C Data Register (PORTC)
$YFFA42	Not Used	Not Used
$YFFA44	Chip-Select Pin Assignment Register 0 (CSPAR0)	
$YFFA46	Chip-Select Pin Assignment Register 1 (CSPAR1)	
$YFFA48	Chip-Select Base Address Register Boot (CSBARBT)	
$YFFA4A	Chip-Select Option Register Boot (CSORBT)	
	Chip-Select Base Address Register 0 (CSBAR0)	

PADA[7:0] — Port ADA Data Pins

A read of PADA[7:0] returns the logic level of the port ADA pins. If an input is not at an appropriate logic level (that is, outside the defined levels), the read is indeterminate. Use of a port ADA pin for digital input does not preclude its simultaneous use as an analog input.

D.5.4 ADC Control Register 0

ADCTL0 — ADC Control Register 0 **$YFF70A**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED								RES10	STS[1:0]		PRS[4:0]				
RESET:								0	0	0	0	0	0	1	1

ADCTL0 is used to select 8- or 10-bit conversions, sample time, and ADC clock frequency. Writes to it have immediate effect.

RES10 — 10-Bit Resolution

0 = 8-bit conversion
1 = 10-bit conversion

Conversion results are appropriately aligned in result registers to reflect the number of bits.

STS[1:0] — Sample Time Selection

Total conversion time is the sum of initial sample time, transfer time, final sample time, and resolution time. Initial sample time is fixed at two ADC clocks. Transfer time is fixed at two ADC clocks. Resolution time is fixed at ten ADC clocks for an 8-bit conversion and twelve ADC clocks for a 10-bit conversion. Final sample time is determined by the STS[1:0] field. Refer to **Table D-26**.

Table D-26 Sample Time Selection

STS[1:0]	Sample Time
00	2 ADC Clock Periods
01	4 ADC Clock Periods
10	8 ADC Clock Periods
11	16 ADC Clock Periods

PRS[4:0] — Prescaler Rate Selection

The ADC clock is derived from the system clock by a programmable prescaler. ADC clock period is determined by the value of the PRS field in ADCTL0. The prescaler has two stages. The first stage is a 5-bit modulus counter. It divides the system clock by any value from two to 32 (PRS[4:0] = %00000 to %11111). The second stage is a divide-by-two circuit. Refer to **Table D-27**.

```
***** SRAM MODULE REGISTERS *****
RAMMCR  EQU $FB00   ;RAM MODULE CONFIGURATION REGISTER
RAMTST  EQU $FB02   ;RAM TEST REGISTER
RAMBAH  EQU $FB04   ;RAM BASE ADDRESS HIGH REGISTER
RAMBAL  EQU $FB06   ;RAM BASE ADDRESS LOW REGISTER
***** MRM MODULE REGISTERS *****
MRMCR   EQU $F820   ;MASKED ROM MODULE CONFIGURATION REGISTER
ROMBAH  EQU $F824   ;ROM ARRAY BASE ADDRESS REGISTER HIGH
ROMBAL  EQU $F826   ;ROM ARRAY BASE ADDRESS REGISTER LOW
SIGHI   EQU $F828   ;SIGNATURE REGISTER HIGH
SIGLO   EQU $F82A   ;SIGNATURE REGISTER LOW
ROMBS0  EQU $F830   ;ROM BOOTSTRAP WORD 0
ROMBS1  EQU $F832   ;ROM BOOTSTRAP WORD 1
ROMBS2  EQU $F834   ;ROM BOOTSTRAP WORD 2
ROMBS3  EQU $F836   ;ROM BOOTSTRAP WORD 3
***** QSM MODULE REGISTERS *****
QMCR    EQU $FC00   ;QSM MODULE CONFIGURATION REGISTER
QTEST   EQU $FC02   ;QSM TEST REGISTER
QILR    EQU $FC04   ;QSM INTERRUPT LEVELS REGISTER
QIVR    EQU $FC05   ;QSM INTERRUPT VECTOR REGISTER
SCCR0   EQU $FC08   ;SCI CONTROL REGISTER 0
SCCR1   EQU $FC0A   ;SCI CONTROL REGISTER 1
SCSR    EQU $FC0C   ;SCI STATUS REGISTER
SCDR    EQU $FC0E   ;SCI DATA REGISTER (FULL WORD, NOT LAST 8 BITS)
QPDR    EQU $FC15   ;QSM PORT DATA REGISTER
QPAR    EQU $FC16   ;QSM PIN ASSIGNMENT REGISTER
QDDR    EQU $FC17   ;QSM DATA DIRECTION REGISTER
SPCR0   EQU $FC18   ;QSPI CONTROL REGISTER 0
SPCR1   EQU $FC1A   ;QSPI CONTROL REGISTER 1
SPCR2   EQU $FC1C   ;QSPI CONTROL REGISTER 2
SPCR3   EQU $FC1E   ;QSPI CONTROL REGISTER 3
SPSR    EQU $FC1F   ;QSPI STATUS REGISTER
RR0     EQU $FD00   ;SPI REC.RAM 0
RR1     EQU $FD02   ;SPI REC.RAM 1
RR2     EQU $FD04   ;SPI REC.RAM 2
RR3     EQU $FD06   ;SPI REC.RAM 3
RR4     EQU $FD08   ;SPI REC.RAM 4
RR5     EQU $FD0A   ;SPI REC.RAM 5
RR6     EQU $FD0C   ;SPI REC.RAM 6
RR7     EQU $FD0E   ;SPI REC.RAM 7
RR8     EQU $FD00   ;SPI REC.RAM 8
RR9     EQU $FD02   ;SPI REC.RAM 9
RRA     EQU $FD04   ;SPI REC.RAM A
RRB     EQU $FD06   ;SPI REC.RAM B
RRC     EQU $FD08   ;SPI REC.RAM C
RRD     EQU $FD0A   ;SPI REC.RAM D
RRE     EQU $FD0C   ;SPI REC.RAM E
RRF     EQU $FD0E   ;SPI REC.RAM F
TR0     EQU $FD20   ;SPI TXD.RAM 0
TR1     EQU $FD22   ;SPI TXD.RAM 1
TR2     EQU $FD24   ;SPI TXD.RAM 2
TR3     EQU $FD26   ;SPI TXD.RAM 3
TR4     EQU $FD28   ;SPI TXD.RAM 4
TR5     EQU $FD2A   ;SPI TXD.RAM 5
TR6     EQU $FD2C   ;SPI TXD.RAM 6
```