M68HC08 Microcontroller

The MC68HC908GP32

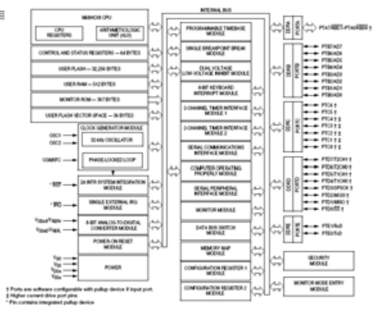
Babak Kia
Adjunct Professor
Boston University
College of Engineering
Email: bkia -at- bu.edu

ENG SC757 - Advanced Microprocessor Design

General Description

- The GP32 is a member of the low-cost, high performance CPU08 family of Microcontrollers
- It operates on 8 MHz internal bus frequency
- 32 Kbytes of on board flash with in-circuit programming capability and security
- 512 bytes of RAM
- Low-power and fully static design
- Peripherals such as SPI, SCI, ADC, two Timer Channels each with input capture, output compare, and PWM
- Up to 33 general purpose I/O pins
- 8-bit Keyboard wakeup port

MCU Block Diagram

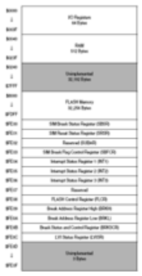

INTERNAL BUS
M68HC08 CPU
CPU REGISTERS
ARITHMETIC/LOGIC UNIT (ALU)
CONTROL AND STATUS REGISTERS — 64 BYTES
USER FLASH — 32,256 BYTES
USER RAM — 512 BYTES
MONITOR ROM — 307 BYTES
USER FLASH VECTOR SPACE — 36 BYTES
CLOCK GENERATOR MODULE
32-kHz OSCILLATOR
PHASE-LOCKED LOOP
OSC1
OSC2
CGMXFC
SIM MODULE
SINGLE EXTERNAL IRQ MODULE
8-BIT ANALOG-TO-DIGITAL CONVERTER MODULE
POWER-ON RESET MODULE
POWER
PROGRAMMABLE TIMEBASE MODULE
SINGLE BREAKPOINT BREAK MODULE
DUAL VOLTAGE LOW-VOLTAGE INHIBIT MODULE
8-BIT KEYBOARD INTERRUPT MODULE
2-CHANNEL TIMER INTERFACE MODULE 1
2-CHANNEL TIMER INTERFACE MODULE 2
SERIAL COMMUNICATIONS INTERFACE MODULE
COMPUTER OPERATING PROPERLY MODULE
SERIAL PERIPHERAL INTERFACE MODULE
MONITOR MODULE
DATA BUS SWITCH MODULE
MEMORY MAP MODULE
CONFIGURATION REGISTER 1 MODULE
CONFIGURATION REGISTER 2 MODULE
DDRA PORTA
DDRB PORTB
DDRC PORTC
DDRD PORTD
DDRE PORTE
SECURITY MODULE
MONITOR MODE ENTRY MODULE

† Ports are software configurable with pullup device if input port.
‡ Higher current drive port pins
* Pin contains integrated pullup device

Figure 1-1. MCU Block Diagram

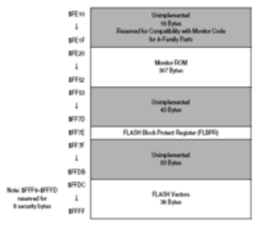
Memory Map

- The CPU08 can address up to 64 Kbytes of memory space
- The memory map to the right shows
  - 32,256 bytes of Flash
  - 512 bytes of RAM
  - 36 bytes of user defined vectors
  - 307 bytes of Monitor ROM
- Addressing unimplemented memory regions can cause an illegal address reset

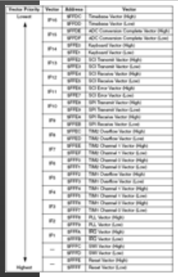

$0000 ↓ $003F
I/O Registers 64 Bytes
$0040 ↓ $023F
RAM 512 Bytes
$0240 ↓ $7FFF
Unimplemented 32,192 Bytes
$8000 ↓ $FDFF
FLASH Memory 32,256 Bytes
$FE00 SIM Break Status Register (SBSR)
$FE01 SIM Reset Status Register (SRSR)
$FE02 Reserved (SUBAR)
$FE03 SIM Break Flag Control Register (SBFCR)
$FE04 Interrupt Status Register 1 (INT1)
$FE05 Interrupt Status Register 2 (INT2)
$FE06 Interrupt Status Register 3 (INT3)
$FE07 Reserved
$FE08 FLASH Control Register (FLCR)
$FE09 Break Address Register High (BRKH)
$FE0A Break Address Register Low (BRKL)
$FE0B Break Status and Control Register (BRKSCR)
$FE0C LVI Status Register (LVISR)
$FE0D ↓ $FE0F
Unimplemented 3 Bytes

Memory Map (Cont.)

- The CPU08 can address up to 64 Kbytes of memory space
- The memory map to the right shows
  - 32,256 bytes of Flash
  - 512 bytes of RAM
  - 36 bytes of user defined vectors
  - 307 bytes of Monitor ROM
- Addressing unimplemented memory regions can cause an illegal address reset


$FE10 ↓ $FE1F
Unimplemented 16 Bytes Reserved for Compatibility with Monitor Code for A-Family Parts
$FE20 ↓ $FF52
Monitor ROM 307 Bytes
$FF53 ↓ $FF7D
Unimplemented 43 Bytes
$FF7E FLASH Block Protect Register (FLBPR)
$FF7F ↓ $FFDB
Unimplemented 93 Bytes
$FFDC ↓ $FFFF
FLASH Vectors 36 Bytes
Note: $FFF6–$FFFD reserved for 8 security bytes

Vector Address



- The Vector Address table shows the addresses for each interrupt source
- Each interrupt source can itself be caused by any number of events which needs to be identified within the Interrupt Service Routine
- Interrupts are prioritized based on their vector address location

Low Power Modes

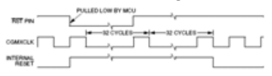
- The MCU provides two low-power modes, the Wait Mode and the Stop Mode. Both are entered as a results of instruction execution
  - Wait Mode: The WAIT instruction puts the MCU in a low-power standby mode, CPU clock is disabled but the Bus clock continues to run
  - Stop Mode: The STOP instruction puts the MCU in a stop mode. Both CPU and Bus clocks are disabled

Exiting Low Power Modes

- A number of events restart the CPU clock and exit the MCU from a Wait Mode
  - External Reset
  - External Interrupt (IRQ#)
  - Keyboard Interrupt
  - Timer Interrupt
  - SPI or SCI Interrupts
- A Stop Mode is exited through one of the following events
  - External Reset
  - External Interrupt (IRQ#)
  - Timebase Module Interrupt (TBM), which allows the TBM module to generate a periodic wakeup signal

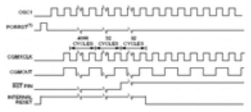
Reset

- Resets are intended to start up the processor from a known startup state
- The reset vector is fetched from memory location $FFFE:FFFF
- There are two types of resets, an external reset where the RST# pin is pulled low, or an internal reset
- In case of an internal reset, the MCU pulls the RST# pin low to allow for resetting of external devices:



Reset

- Internal Resets have several sources
  - Power-on Reset (POR)
  - Computer Operating Properly (COP)
  - Low-Power Reset Circuit
  - Illegal Opcode Reset
  - Illegal Address Reset
- A POR is an internal reset caused by a positive transition on the V_{DD} pin

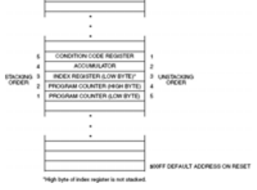


Interrupts

- An interrupt is an external event which temporarily changes the execution path
- At the end of each instruction, the CPU checks all pending interrupts, if the I bit is set
- If more than one interrupt is pending when the instruction is done, the highest priority interrupt is serviced first
- An interrupt does not stop the current instruction from execution, but will change execution path once the current operation is finished

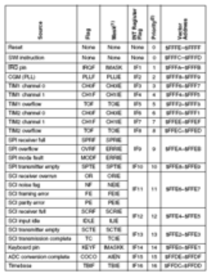
Interrupts

- Upon an interrupt, the CPU registers are saved onto the stack in the following order
- Once an interrupt occurs, the processor sets the I mask in order to prevent other interrupts from occurring
- At the end of the interrupt service routine, the RTI instruction will restore the CPU registers in the reverse order


5 CONDITION CODE REGISTER 1
4 ACCUMULATOR 2
STACKING ORDER
3 INDEX REGISTER (LOW BYTE)* 3
UNSTACKING ORDER
2 PROGRAM COUNTER (HIGH BYTE) 4
1 PROGRAM COUNTER (LOW BYTE) 5
$00FF DEFAULT ADDRESS ON RESET
*High byte of index register is not stacked.

Interrupt Processing

- In addition to the I-mask, each interrupt source has its own mask bits
- Reset and SWI instruction cannot be masked
- A software interrupt (SWI) pushes the value of PC onto the stack. It *does not* push the value of PC-1 which is the case for a hardware interrupt



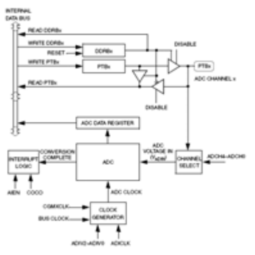
Source	Flag	Mask[1]	INT Register Flag	Priority[2]	Vector Address
Reset	None	None	None	0	$FFFE–$FFFF
SWI instruction	None	None	None	0	$FFFC–$FFFD
IRQ pin	IRQF	IMASK	IF1	1	$FFFA–$FFFB
CGM (PLL)	PLLF	PLLIE	IF2	2	$FFF8–$FFF9
TIM1 channel 0	CH0F	CH0IE	IF3	3	$FFF6–$FFF7
TIM1 channel 1	CH1F	CH1IE	IF4	4	$FFF4–$FFF5
TIM1 overflow	TOF	TOIE	IF5	5	$FFF2–$FFF3
TIM2 channel 0	CH0F	CH0IE	IF6	6	$FFF0–$FFF1
TIM2 channel 1	CH1F	CH1IE	IF7	7	$FFEE–$FFEF
TIM2 overflow	TOF	TOIE	IF8	8	$FFEC–$FFED
SPI receiver full	SPRF	SPRIE	IF9	9	$FFEA–$FFEB
SPI overflow	OVRF	ERRIE			
SPI mode fault	MODF	ERRIE			
SPI transmitter empty	SPTE	SPTIE	IF10	10	$FFE8–$FFE9
SCI receiver overrun	OR	ORIE	IF11	11	$FFE6–$FFE7
SCI noise flag	NF	NEIE			
SCI framing error	FE	FEIE			
SCI parity error	PE	PEIE			
SCI receiver full	SCRF	SCRIE	IF12	12	$FFE4–$FFE5
SCI input idle	IDLE	ILIE			
SCI transmitter empty	SCTE	SCTIE	IF13	13	$FFE2–$FFE3
SCI transmission complete	TC	TCIE			
Keyboard pin	KEYF	IMASKK	IF14	14	$FFE0–$FFE1
ADC conversion complete	COCO	AIEN	IF15	15	$FFDE–$FFDF
Timebase	TBIF	TBIE	IF16	16	$FFDC–$FFDD

Analog-to-Digital Converter

- The ADC provides 8 bit resolution for converting an analog value at the pin into a digital format
- It has 8 channels with multiplexed inputs
- Performs either a single or a continuous conversion
- Provides a conversion complete flag, or a conversion complete interrupt

ADC

- The general-purpose I/O pins on Port B share pin space with the ADC module
- Once configured, the ADC forces the pins to act as inputs of the ADC circuitry and therefore bypass Port B functionality
- Writes to Port B will have no functionality when in ADC mode

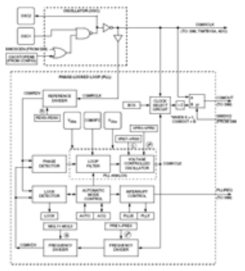


ADC

- The ADC module works in the following fashion
  - When the input voltage at an ADC channel equals V_{REFH}, the ADC converts the voltage to $FF
  - If the input equals V_{REFL}, the ADC coverts it to $00
  - If the input is between V_{REFL} and V_{REFH}, the ADC performs a straight-line linear conversion

Clock Generation Module

- The Clock Generation Module generates the crystal clock signal, CGMXCLK, which operates at the frequency of the crystal
- It also generates the base clock, CGMOUT, which is then used by the SIM to derive the system clocks, including the bus clock which runs at CGMOUT/2

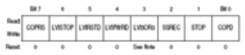


Phase-Locked Loop (PLL)

- The PLL is a frequency generator which operates in either an acquisition mode or a tracking mode, depending on the desired accuracy
- In acquisition mode the PLL filter makes large frequency corrections, which is used when the PLL starts up, or has suffered a severe noise hit
- In tracking mode, the filter makes small corrections to the frequency
- The PLL generates an 8 MHz bus frequency using a 32 KHz crystal

Configuration Register

- The Configuration Register (CONFIG) is used to setup various CPU parameters
- The CONFIG register can only be written once after each reset
- Since this configuration affects the operations of the CPU, it is recommended that it be written immediately after reset

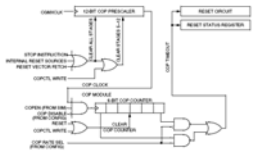


	Bit 7	6	5	4	3	2	1	Bit 0
Read / Write	COPRS	LVISTOP	LVIRSTD	LVIPWRD	LVI5OR3	SSREC	STOP	COPD
Reset	0	0	0	0	See Note	0	0	0

- The CONFIG register's more important bits are COPRS and COPD
  - COPRS = 1 selects a rate of 2^{13}-2^{4} CGMXCLK cycles
  - COPRS = 0 selects a rate of 2^{18}-2^{4} CGMXCLK cycles
  - COPD enables (0), or disables (1) the watchdog timer

COP

- The COP watchdog module generates a system reset if it is enabled and its free running counter is allowed to overflow
- System software can prevent a system reset by writing any value to the memory location $FFFF periodically
- Resetting the COP timer must be done in the program itself, and not as part of an Interrupt Service Routine!

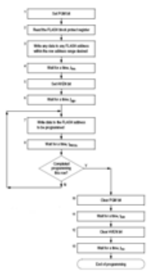


Flash

- The flash on the GP32 is an array of 32,256 bytes, with an additional 36 bytes of user interrupt vectors, and one byte of block protection
- Erasing and Programming of flash traditionally require an external power supply with supply voltages exceeding that of V_{DD}
- However, for ease of use, the flash on the CPU08 family of microcontrollers can be erased and programmed using an internal charge pump
- This reduces need for external circuitry and different power supplies

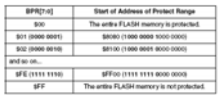
Flash

- This algorithm outlines the steps necessary for programming a row (64 bytes) of Flash memory


Set PGM bit
Read the FLASH block protect register
Write any data to any FLASH address within the row address range desired
Wait for a time, t_{nvs}
Set HVEN bit
Wait for a time, t_{pgs}
Write data to the FLASH address to be programmed
Wait for a time, t_{PROG}
Completed programming this row?
Y
N
Clear PGM bit
Wait for a time, t_{nvh}
Clear HVEN bit
Wait for a time, t_{rcv}
End of programming

Flash

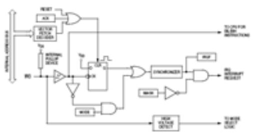
- The Flash provides a means of protecting blocks from unintentional erase and modification. This is performed through setting the FLBPR register to cover the range of flash which needs to be protected.
- The following table lists examples of protecting flash from accidental modification and erasure



BPR[7:0]	Start of Address of Protect Range
$00	The entire FLASH memory is protected.
$01 (0000 0001)	$8080 (1000 0000 1000 0000)
$02 (0000 0010)	$8100 (1000 0001 0000 0000)
and so on...	
$FE (1111 1110)	$FF00 (1111 1111 0000 0000)
$FF	The entire FLASH memory is not protected.

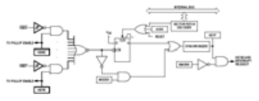
External Interrupt (IRQ)

- The External Interrupt Request Pin provides an external means of interrupting the CPU
- Some of its features are:
  - Programmable edge-only or edge and level interrupt sensitivity
  - Hysteresis buffer
  - Automatic Interrupt Acknowledge
  - Internal Pullup resistor



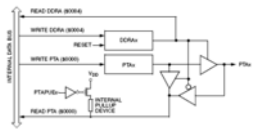
Keyboard Interrupt (KBI)

- The Keyboard Interrupt module (KBI) provides 8 independently maskable external interrupt sources to the processor
- Some of its features are
  - Programmable edge-only or edge and level interrupt sensitivity
  - Hysteresis buffer
  - Exit from low-power modes



Input/Output (I/O) Ports

- The following is a diagram of the Port A I/O Circuit
- All Port A, C, and D pins have software configurable pullup circuitry when configured as inputs
- The pullup circuitry is automatically disabled when the port is configured as an output
- Ports have two registers, one which reads/writes values to the port pins, and the other is the Data Direction Register (DDR) determining the I/O mode of the port pins



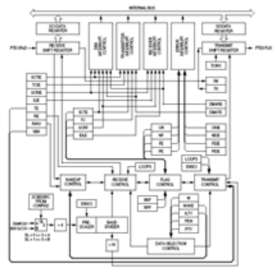
Random Access Memory (RAM)

- The GP32 microcontroller provides 512 bytes of RAM from location $40 - $23F
- RAM is important not only for maintaining variables, but also for providing stack
- Even though the 16-bit CPU08 Stack Pointer can be initialized to point to anywhere within a 64Kbyte memory space, only pointing it to the RAM location guarantees correct operation
- Stack Pointer is set to $FF upon reset, but can be programmed to point to $23F
- This frees up the page zero RAM locations for the full benefit of direct addressing mode instructions (which, as a reminder, only target page zero RAM)

Serial Communications Interface

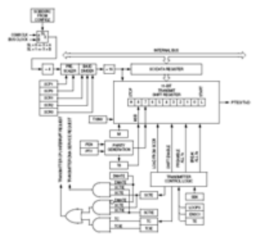
- The Serial Communications Interface provides a means for high-speed synchronous communication with external peripherals
- SCI features on the GP32 include
  - Full duplex operation
  - Standard mark-space Non-Return-to-Zero (NRZ) format
  - 32 programmable baud rates
  - Separately enabled transmitter and receiver
  - Receiver and transmitter interrupts
  - Programmable transmitter output polarity

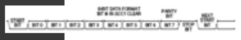
SCI



SCI Transmitter

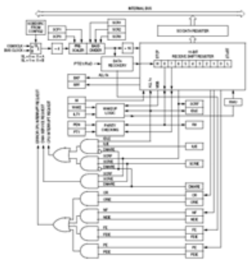
- Transmitter needs to be enabled through
  - Enabling SCI (ENSCI)
  - Enabling Transmitter (TE)
  - Clearing the SCTE
- For each consecutive transmission, SCTE must be cleared
- The following is the 8-bit data format used for transmitting a byte





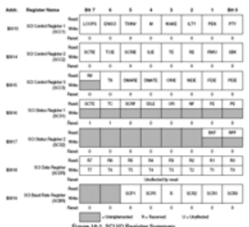
SCI Receiver

- The SCI Receiver automatically generates interrupts (if enabled) for a number of error conditions
  - Overrun Error
  - Noise Error
  - Framing Error
  - Parity Error
- It also has the means for idle line detection and receiver wake-up functionality



SCI Register Summary

- SCC1
  - Enables the SCI
  - Controls Character length
  - Enables Parity function
  - Enables Parity type
- SCC2
  - Enables TXD and RXD Interrupts
  - Enables Receiver and Transmitter
- SCC3
  - Other interrupt sources such as parity and noise



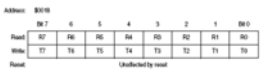
Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0013	SCI Control Register 1 (SCC1)	Read/Write	LOOPS	ENSCI	TXINV	M	WAKE	ILTY	PEN	PTY
		Reset	0	0	0	0	0	0	0	0
$0014	SCI Control Register 2 (SCC2)	Read/Write	SCTIE	TCIE	SCRIE	ILIE	TE	RE	RWU	SBK
		Reset	0	0	0	0	0	0	0	0
$0015	SCI Control Register 3 (SCC3)	Read	R8	T8	DMARE	DMATE	ORIE	NEIE	FEIE	PEIE
		Write	▨							
		Reset	U	U	0	0	0	0	0	0
$0016	SCI Status Register 1 (SCS1)	Read	SCTE	TC	SCRF	IDLE	OR	NF	FE	PE
		Write	▨	▨	▨	▨	▨	▨	▨	▨
		Reset	1	1	0	0	0	0	0	0
$0017	SCI Status Register 2 (SCS2)	Read	▨	▨	▨	▨	▨	▨	BKF	RPF
		Write	▨	▨	▨	▨	▨	▨	▨	▨
		Reset	0	0	0	0	0	0	0	0
$0018	SCI Data Register (SCDR)	Read	R7	R6	R5	R4	R3	R2	R1	R0
		Write	T7	T6	T5	T4	T3	T2	T1	T0
		Reset	Unaffected by reset							
$0019	SCI Baud Rate Register (SCBR)	Read/Write	▨	▨	SCP1	SCP0	R	SCR2	SCR1	SCR0
		Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented R = Reserved U = Unaffected

Figure 18-2. SCI I/O Register Summary

SCI Data and Baud Registers

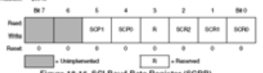
- The SCI Data Register (SCDR) is used to read/write data from SCI module



Address: $0018

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	R7	R6	R5	R4	R3	R2	R1	R0
Write:	T7	T6	T5	T4	T3	T2	T1	T0
Reset:	Unaffected by reset							

- The SCI Baud Rate Register (SCBR) sets the baud rate for both the transmitter and the receiver



Address: $0019

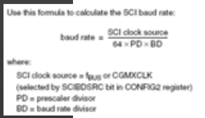
	Bit 7	6	5	4	3	2	1	Bit 0
Read:/Write:	▨	▨	SCP1	SCP0	R	SCR2	SCR1	SCR0
Reset:	0	0	0	0	0	0	0	0

▨ = Unimplemented R = Reserved

Figure 18-16. SCI Baud Rate Register (SCBR)

Setting Baud Rate

- The SCI module can be configured with a number of prescaler and baud rate divisors, the following table outlines baud rates with different prescalers

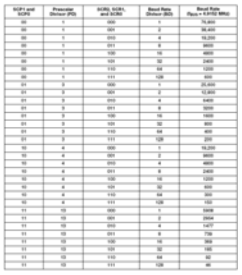


Use this formula to calculate the SCI baud rate:

$$\text{baud rate} = \frac{\text{SCI clock source}}{64 \times PD \times BD}$$

where:

SCI clock source = f_{BUS} or CGMXCLK
(selected by SCIBDSRC bit in CONFIG2 register)
PD = prescaler divisor
BD = baud rate divisor



SCP1 and SCP0	Prescaler Divisor (PD)	SCR2, SCR1, and SCR0	Baud Rate Divisor (BD)	Baud Rate (f_{BUS} = 4.9152 MHz)
00	1	000	1	76,800
00	1	001	2	38,400
00	1	010	4	19,200
00	1	011	8	9600
00	1	100	16	4800
00	1	101	32	2400
00	1	110	64	1200
00	1	111	128	600
01	3	000	1	25,600
01	3	001	2	12,800
01	3	010	4	6400
01	3	011	8	3200
01	3	100	16	1600
01	3	101	32	800
01	3	110	64	400
01	3	111	128	200
10	4	000	1	19,200
10	4	001	2	9600
10	4	010	4	4800
10	4	011	8	2400
10	4	100	16	1200
10	4	101	32	600
10	4	110	64	300
10	4	111	128	150
11	13	000	1	5908
11	13	001	2	2954
11	13	010	4	1477
11	13	011	8	739
11	13	100	16	369
11	13	101	32	185
11	13	110	64	92
11	13	111	128	46

Setting up SCI to transmit

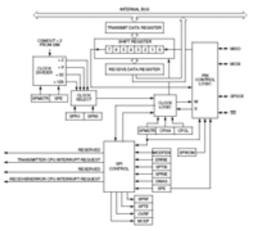
- The following steps are required to configure the SCI module for transmission
  1. Set the desired Baud Rate, Interrupt Sources, Parity, Stop bits, and character length
  2. Enable SCI by writing a logic 1 to the ENSCI bit of SCC1
  3. Enable the transmitter by writing a logic 1 to the Transmitter Enable (TE) bit of SCC2
  4. Clear the SCI Transmitter Empty flag by first reading SCS1, and then writing to SCDR
  5. Repeat the last step for each subsequent character transmission

System Integration Module

- The SIM module works in conjunction with the CPU to control major functionality of the MCU
- Provides bus clock generation and control for the CPU and its peripherals, including
  - Stop/wait/reset/break entry and recovery
  - Internal clock control
- Controls master reset control, including POR and COP watchdog
- Provides Interrupt control and management
  - Acknowledge
  - Arbitration
  - Vector address generation

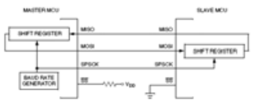
Serial Peripheral Interface (SPI)

- The SPI allows for synchronous full-duplex communication amongst peripherals
- Amongst its many features are
  - Full-duplex operation
  - Master/Slave modes
  - Double-buffered operation
  - Serial clock with programmable polarity and phase
  - Receiver-full and Transmitter-empty interrupts
  - Programmable Wired-OR mode
  - I²C (inter-integrated circuit) compatibility



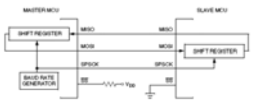
SPI - Master Mode

- Only a Master SPI module can initiate transmissions
- The Master also controls the shift register and baud rate of the slave module through the SPSCK clock pin
- Data shifts out from the Master on the MOSI pin, and at the same time data shifts in from the Slave on the MISO pin



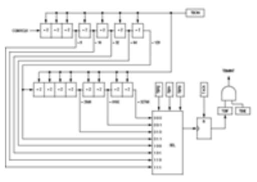
SPI – Slave Mode

- Before data is accepted by the Slave SPI module, the Slave Select (SS#) pin must be a logic 0
- This allows for multiple slave devices to share the same connection through a Wired-OR topology
- The SS# pin must remain low throughout the transmission



Timebase Module (TBM)

- The TBM generates periodic timer interrupts at a user selectable rate
- It provides a software programmable interrupt at 1, 4, 16, 256, 512, 1924, 2048, and 4906 Hz using an external 32.768 KHz crystal
- This module allows for periodic interrupts to wakeup the MCU from a STOP mode

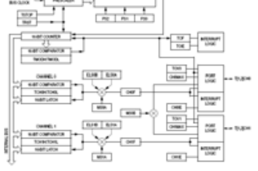


Timer Interface Module (TIM)

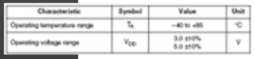
- The Timer Interface Module is a two channel timer which provides a timing reference with input-capture, output-compare, and Pulse Width Modulation features
- Features of the TIM include
  - Rising-edge, falling-edge, or any-edge input capture
  - Set, clear, and toggle output-compare functionality
  - Buffered and unbuffered PWM signal generation
  - Free-running or modulo up-count operation
  - Toggle any channel pin on overflow

TIM

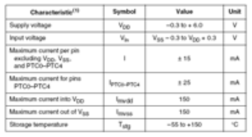
- Input-capture enables the MCU to capture the time an external event has occurred
- With an output-compare, the MCU can generate a periodic pulse with programmable duration, frequency, and polarity



Electrical Specifications

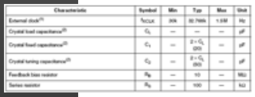


Characteristic	Symbol	Value	Unit
Operating temperature range	T_A	−40 to +85	°C
Operating voltage range	V_{DD}	3.0 ±10% 5.0 ±10%	V

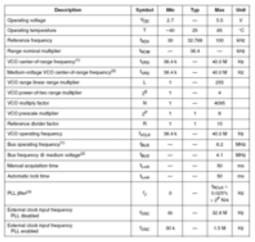


Characteristic[1]	Symbol	Value	Unit
Supply voltage	V_{DD}	−0.3 to + 6.0	V
Input voltage	V_{in}	V_{SS} − 0.3 to V_{DD} + 0.3	V
Maximum current per pin excluding V_{DD}, V_{SS}, and PTC0–PTC4	I	± 15	mA
Maximum current for pins PTC0–PTC4	$I_{PTC0-PTC4}$	± 25	mA
Maximum current into V_{DD}	I_{mvdd}	150	mA
Maximum current out of V_{SS}	I_{mvss}	150	mA
Storage temperature	T_{stg}	−55 to +150	°C

CGM Specifications

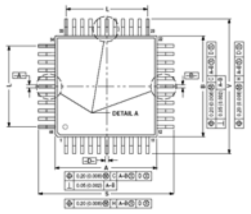


Characteristic	Symbol	Min	Typ	Max	Unit
External clock[1]	f_{XCLK}	30k	32.768k	1.5M	Hz
Crystal load capacitance[2]	C_L	—	—	—	pF
Crystal fixed capacitance[2]	C_1	—	2 × C_L (20)	—	pF
Crystal tuning capacitance[2]	C_2	—	2 × C_L (20)	—	pF
Feedback bias resistor	R_B	—	10	—	MΩ
Series resistor	R_S	—	100	—	kΩ



Description	Symbol	Min	Typ	Max	Unit
Operating voltage	V_{DD}	2.7	—	5.5	V
Operating temperature	T	−40	25	85	°C
Reference frequency	f_{RDV}	30	32.768	100	kHz
Range nominal multiplier	f_{NOM}	—	38.4	—	kHz
VCO center-of-range frequency[1]	f_{VRS}	38.4 k	—	40.0 M	Hz
Medium-voltage VCO center-of-range frequency[2]	f_{VRS}	38.4 k	—	40.0 M	Hz
VCO range linear range multiplier	L	1	—	255	
VCO power-of-two range multiplier	2^E	1	—	4	
VCO multiply factor	N	1	—	4095	
VCO prescale multiplier	2^P	1	1	8	
Reference divider factor	R	1	1	15	
VCO operating frequency	f_{VCLK}	38.4 k	—	40.0 M	Hz
Bus operating frequency[1]	f_{BUS}	—	—	8.2	MHz
Bus frequency @ medium voltage[2]	f_{BUS}	—	—	4.1	MHz
Manual acquisition time	t_{Lock}	—	—	50	ms
Automatic lock time	t_{Lock}	—	—	50	ms
PLL jitter[3]	f_J	0	—	f_{RCLK} × 0.025% × 2^P N/4	Hz
External clock input frequency PLL disabled	f_{OSC}	dc	—	32.8 M	Hz
External clock input frequency PLL enabled	f_{OSC}	30 k	—	1.5 M	Hz

Mechanical Specifications


L
DETAIL A
A
B
D
S
0.20 (0.008)
0.05 (0.002)