M68HC08 Microcontroller

The MC68HC908GP32

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

M68HC08 Microcontroller

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
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.

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.

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 result 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 wake-up 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 VDD 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

- 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
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

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
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 converts it to $00$.
- If the input is between $V_{REFL}$ and $V_{REFH}$, the ADC performs a straight-line linear conversion.

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 $CGMOUT/2$.

The Phase-Locked Loop (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.
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. The CONFIG register’s more important bits are COPRS and COPD:

- COPRS = 1 selects a rate of $2^{13-24}$ CGMXCLK cycles
- COPRS = 0 selects a rate of $2^{18-24}$ CGMXCLK cycles
- COPD enables (0), or disables (1) the watchdog timer

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!

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

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:

<table>
<thead>
<tr>
<th>RBPX</th>
<th>Start of Address of Flash Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No, the entire memory is protected</td>
</tr>
<tr>
<td>0x2000 0000</td>
<td>0x20000000 to 0x20000000</td>
</tr>
<tr>
<td>0x2500 0000</td>
<td>0x25000000 to 0x25000000</td>
</tr>
<tr>
<td>0x2B00 0000</td>
<td>0x2B000000 to 0x2B000000</td>
</tr>
<tr>
<td>0x3200 0000</td>
<td>0x32000000 to 0x32000000</td>
</tr>
<tr>
<td>0x3700 0000</td>
<td>0x37000000 to 0x37000000</td>
</tr>
<tr>
<td>0x4000 0000</td>
<td>0x40000000 to 0x40000000</td>
</tr>
</tbody>
</table>

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).
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 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
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

SCI Data and Baud Registers

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

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

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

![Baud Rate Table]

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
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 wake up 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

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

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Symbol</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supply voltage</td>
<td>Vcc</td>
<td>5.0</td>
<td>V</td>
</tr>
<tr>
<td>Output current</td>
<td>Iout</td>
<td>200</td>
<td>mA</td>
</tr>
<tr>
<td>Overvoltage rating</td>
<td>Vbr</td>
<td>6.0</td>
<td>V</td>
</tr>
</tbody>
</table>

CGM Specifications

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum operating temperature</td>
<td>Tmax</td>
<td>150</td>
<td>°C</td>
</tr>
<tr>
<td>Junction temperature</td>
<td>Tj</td>
<td>125</td>
<td>°C</td>
</tr>
<tr>
<td>Current density</td>
<td>Is</td>
<td>1.0</td>
<td>A/mm²</td>
</tr>
</tbody>
</table>

Mechanical Specifications

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Package size</td>
<td>10.0</td>
<td>mm</td>
</tr>
<tr>
<td>Pin count</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>Lead spacing</td>
<td>1.27</td>
<td>mm</td>
</tr>
</tbody>
</table>