• Using the MC9S12 in Expanded Mode – External Ports
• S12CPUV2 Reference Manual
• Multiplexed External Bus Interface (MEBI) Module V3 Block User Guide
- Computer with N bit address bus can access $2^N$ bytes of data

- Computer with M bit data bus can access M bits of data in one memory cycle

- Value on address bus tells memory which location computer wants to read (write)

- Control lines tell memory when computer wants to read (write) data, and if access is read or write
Address, Data and Control Buses

• A microprocessor system uses address, data and control buses to communicate with external memory and memory-mapped peripherals

  • The address bus determines which memory location to access

  • The control bus specifies whether the memory cycle is a read (into microprocessor) or a write cycle (out of microprocessor), and specifies timing information for the cycle

  • The data bus contains the data being transferred during the memory cycle

• For example, consider the following simple MC9S12 program, which continuously increments the contents of address 0x0400:

  org 0x2000
  loop: inc 0x0400
  bra loop

– The program is stored in memory starting at memory location 0x2000

– The MC9S12 Program Counter starts at address 0x2000

– The MC9S12 reads the first instruction, inc 0x0400, located in address 0x2000 through 0x2002
– The MC9S12 then reads the contents of memory location 0x0400, takes an internal memory cycle to increment the value, then writes the new value out to address 0x0400

– The MC9S12 then reads the next instruction, bra 0x2000

– The MC9S12 takes one memory cycle to load the program counter with the new value of 0x2000, and to clear its internal pipeline, then reads the instruction at 0x2000 to figure out what to do next
The MC9S12 address, data and control buses

• Note: The following diagram assumes that the MC9S12 accesses one byte at a time

• The MC9S12 actually accesses two bytes (16 bits) at a time, when it can

• What actually occurs on the MC9S12 bus is a little more complicated than what is shown below
The MC9S12 Memory Map

MC9S12 has 16 bit address bus – can access 65536 bytes
  1024 bytes = 1 kB
  65536 bytes = 64 kB

MC9S12 has 16 bit data bus – can access 16 bits (2 bytes) at a time
For example, the instruction \texttt{LDX \$0900}
will read the two bytes at address \$0900 and \$0901

Sometimes MC9S12 only accesses one byte -- e.g., \texttt{LDAA \$0900}
The MC9S12 accesses only the byte at address \$0900

R/W tells memory (or peripheral) if MC9S12 is reading or writing
  R/W high => read
  R/W low => write

E clock tells memory when MC9S12 is reading (writing) –
synchronizes data accesses

LSTRB (together with ADDR0) tells memory if MC9S12 is
accessing one or two bytes

• The MC9S12 has address regions occupied by internal memory
  and peripherals

• A diagram showing which address regions are used is called a
  memory map
• Here is a memory map of the MC9S12DP256 with no added memory or peripherals

![Memory Map](image)

**The Expanded MC9S12 Memory Map**

• We will add external peripherals to the MC9S12

• First, we will disable the Flash EEPROM at address 0x4000 through 0x7FFF (which we are not using anyway)

• Here is a memory map of the MC9S12DP256 with the peripherals we will add
• The peripherals will be put at 0x4000 and 0x4001

<table>
<thead>
<tr>
<th>Address</th>
<th>Type</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0000</td>
<td>Registers</td>
<td>1 KB</td>
</tr>
<tr>
<td>0x03FF</td>
<td>EEPROM</td>
<td>3 KB</td>
</tr>
<tr>
<td>0x0400</td>
<td>User RAM</td>
<td>11 KB</td>
</tr>
<tr>
<td>0x0FFF</td>
<td>D-Bug 12 RAM</td>
<td>1 KB</td>
</tr>
<tr>
<td>0x1000</td>
<td>Unused Space</td>
<td></td>
</tr>
<tr>
<td>0x3BFF</td>
<td>Use address 0x4000 - 0x4001 for external peripherals</td>
<td></td>
</tr>
<tr>
<td>0x3C00</td>
<td>Banked Flash EEPROM</td>
<td>16 KB</td>
</tr>
<tr>
<td>0x3FFF</td>
<td>D-Bug 12 Flash EEPROM</td>
<td>16 KB</td>
</tr>
<tr>
<td>0x4000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x7FFF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0x8000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0xBFFF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0xC000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0xFFFF</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Simplified MC9S12 Write Cycle

• When the MC9S12 writes data to memory it does the following:
  – It puts the address it wants to write to on the address bus (when E-clock goes low)
  – It puts the data it wants to write onto the data bus
  – It brings the Read/Write (R/W) line low to indicate a write
  – The MC9S12 expects the external device at the given address will latch the data into its registers data on the falling edge of the E-clock
Simplified MC9S12 Read Cycle

- When the MC9S12 reads data from memory it does the following:
  - It puts the address it wants to read from on the address bus (when E-clock goes low)
  - It brings the Read/Write (R/W) line high to indicate a read
  - The MC9S12 expects the external device at the given address will put data on the data bus
  - On the falling edge of the E-clock, the MC9S12 latches the data into its internal register

Example: Read from address 0x5678 & 0x5679
The Real MC9S12DP256 Bus

- Up to now we have been using the MC9S12 in Single Chip Mode
  - In Single Chip Mode the MC9S12 does not have an external address/data bus

- The MC9S12 can be run in Expanded Mode
  - In Expanded Mode the MC9S12 does have an external address/data bus

- Things are a little more complicated on the real MC9S12DP256 bus than shown in the simplified diagrams above:
  - The MC9S12DP256 has a multiplexed address/data bus
  - The MC9S12DP256 sometimes accesses a single byte on a memory cycle, and it sometimes access two bytes on a memory cycle
Figure 1-1 MC9S12DT256 Block Diagram

- 256K Byte Flash EEPROM
- 128K Byte RAM
- 4K Byte EEPROM

Voltage Regulator

Single-wire Background Debug Module

CPU12

PLL

Clock and Reset

Generation Module

System Integration Module (SIM)

Core

Enhanced Capture

Timer

Module to Port Routing

Multiple Address/Data Bus

DDRA

DDRB

Multiple Wired Bus

Internal Logic 2.5V

I/O Driver 5V

PLL 2.5V

A/D Converter 5V & Voltage Regulator Reference

Voltage Regulator 5V & I/O

Multiple Wired Bus

Serials shown in bold are not available on the 8051.
The Multiplexed Address/Data Bus

• The MC9S12DP256 has a limited number of pins it can use

• To have full 16-bit address bus and a full 16-bit data bus the MC9S12DP256 would need to use 32 extra pins (in addition to several pins used for the control bus)

• To save pin count Motorola uses the same set of pins for several purposes

• When put into expanded mode, the MC9S12 uses the pins normally used for Ports A and B for its multiplexed address and data bus
  – When running in expanded mode you can no longer use Ports A and B as general purpose I/O lines

• The MC9S12 uses the same sixteen lines of Ports A and B for both address and data

• When the E-clock is low the sixteen lines AD15-0 are used for address

• When the E-clock is high the sixteen lines AD15-0 are used for data
MC9S12 has 16–bit address and 16–bit data buses

It Requires 35 bits!

**Not enough pins on MC9S12 to allocate 35 pins for buses and pins for all other functions**
Memory Chip Interface

• Memory chips need separate address and data bus
  – Need way to de-multiplex address and data lines from MC9S12

• Memory chips need different control lines than the MC9S12 supplies

• These control lines are:
  – Chip Select – goes low when the MC9S12 is selecting memory chip
  – Write Enable – goes low when the MC9S12 is writing to memory
  – Output/Read Enable – goes low when the MC9S12 is reading from memory
  – High Byte Enable – goes low when the MC9S12 is accessing the High Byte (Odd Address) of memory
  – Low Byte Enable – goes low when the MC9S12 is accessing the Low Byte (Even Address) of memory
The Multiplexed Address/Data Bus

- To talk to memory chip we will need to build a de-multiplexer between the MC9S12 and the memory chip

MCS12 has 16-bit address and 16-bit data buses
Requires 35 bits
Not enough pins on MC9S12 to allocate 35 pins for buses and pins for all other functions

Solution: multiplex address and data buses
MC9S12 uses Ports A and B as multiplexed address/data bus
In expanded mode, you can no longer use Ports A and B for I/O

16-bit Bus: While E low, bus supplies address (from MC9S12)
            While E high, bus supplies data (from MC9S12 on write, from memory on read)