ECE/CS 5780/6780: Embedded System Design

Scott R. Little

Lecture 22: Memory Interfacing
Most embedded systems use only the memory built-in to the microcontroller.

Memory interfacing and bus timing is important to understanding internal microcontroller architecture.

Sometimes internal memory is insufficient, and external memory is needed.

External devices can be interfaced using memory-mapped I/O.
Memory-Mapped I/O
Isolated I/O
Expanded Mode

<table>
<thead>
<tr>
<th>Select</th>
<th>R/W</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Off</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Off</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Write</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Read</td>
</tr>
</tbody>
</table>
Multiplexed Address and Data Lines

MC9S12C32 Bus master

AD15-AD0

E

FlipFlop

Two 74FCT374's

A15-A0

RAM

SelectRAM

Address decoder

I/O

SelectI/O

Address decoder

ROM

SelectROM

Address decoder
Full-Address Decoding

- *Slave* selected only when slave’s address is on the bus.
- Design using the following steps:
  - Write specified address using 0,1,X:
    - 0100,00XX,XXXX,XXXX for 1K RAM at $4000-$43FF
  - Write equation using all 0s and 1s:
    \[
    \text{select} = \overline{A_{15}} \cdot A_{14} \cdot \overline{A_{13}} \cdot \overline{A_{12}} \cdot \overline{A_{11}} \cdot \overline{A_{10}}
    \]
  - Build circuit using gates.
Address Decoder for 1K RAM at $4000-43FF
An Address Decoder for I/O Device at $5500
Minimal-Cost Address Decoding

- Use don’t cares for unspecified addresses to simplify.

Example:

4K RAM $0000 to $0FFF
Input $5000
Output $5001

16K ROM $C000 to $FFFF

Scott R. Little (Lecture 22: Memory Interfacing)
ECE/CS 5780/6780
An Address Decoder

- A15
- A14
- 74HC04
- ROM select
- RAM select
- Output select
- A0
- 74HC11
- Input select
Karnaugh Maps
Special Cases

- Size of the memory is not a power of 2.
  20K RAM with address range $0000$ to $4FFF$
    - $00XX,XXXX,XXXX,XXXX$ Range $0000$ to $3FFF$
    - $0100,XXXX,XXXX,XXXX$ Range $4000$ to $4FFF$

- Start address divided by memory size not an integer.
  32K RAM with address range $2000$ to $9FFF$
    - $001X,XXXX,XXXX,XXXX$ Range $2000$ to $3FFF$
    - $01XX,XXXX,XXXX,XXXX$ Range $4000$ to $7FFF$
    - $100X,XXXX,XXXX,XXXX$ Range $8000$ to $9FFF$
Programmable Address Decoder

\[
\begin{array}{cccccc}
<table>
<thead>
<tr>
<th>\text{In}</th>
<th>\text{Mn}</th>
<th>\text{An}</th>
<th>\text{Vn}</th>
<th>\text{Out}</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
\end{array}
\]
Timing Intervals

\[
\begin{align*}
(\uparrow Y, \downarrow Y) &= (\downarrow A, \uparrow A) + 10 \\
(\uparrow Y, \downarrow Y) &= (\downarrow A, \uparrow A) + [5, 15] \\
(\uparrow Y, \downarrow Y) &= (\downarrow A + [8, 15], \uparrow A + [5, 12])
\end{align*}
\]
Available and Required Time Intervals

Data available DA

Data required DR

\[
DA = (\downarrow G^* + [10, 20], \uparrow G^* + [0, 15])
\]

\[
DA = (\downarrow G^* + 20, \uparrow G^*) \text{ worst-case}
\]

\[
DR = (\uparrow Clk - 30, \uparrow Clk + 5)
\]
# Timing Diagrams

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The input must be valid</td>
<td>The output will be valid</td>
</tr>
<tr>
<td></td>
<td>If the input were to fall</td>
<td>Then the output will fall</td>
</tr>
<tr>
<td></td>
<td>If the input were to rise</td>
<td>Then the output will rise</td>
</tr>
<tr>
<td></td>
<td>Don't care, it will work regardless</td>
<td>Don't know, the output value is indeterminate</td>
</tr>
<tr>
<td></td>
<td>Nonsense</td>
<td>High impedance, tristate, HiZ, Not driven, floating</td>
</tr>
</tbody>
</table>
Read Cycle Circuit

Read cycle circuits

Microcomputer

Data bus

D7-D0

Slave1 data

\( \overline{G_1} \)

Slave2 data

\( \overline{G_2} \)

Slave3 data

\( \overline{G_3} \)
Write Cycle Circuit

Write cycle circuits

Microcomputer

D7-D0

Data bus

Slave1 data

Slave2 data

Slave3 data

C1

C2

C3
Synchronous Bus Timing

Synchronized read cycle

Synchronized write cycle

E

\overline{G_1}

D7-D0

E

C_1

D7-D0

t_{su}

t_h

Scott R. Little (Lecture 22: Memory Interfacing)
Partially Asynchronous Bus Timing
(6809/680x0/x86)
Fully Asynchronous Read Cycle
Fully Asynchronous Write Cycle

By master
I/O

By slave
ACK

REQ

Data

1 -> 2

3 -> 4

XX
Four Types of Control Signals

- Unsynchronized positive logic control signal
- Unsynchronized negative logic control signal
- Synchronized positive logic control signal
- Synchronized negative logic control signal
## MC9S12C32 Modes of Operation

<table>
<thead>
<tr>
<th>BKGD</th>
<th>MODB</th>
<th>MODA</th>
<th>Description</th>
<th>Port A</th>
<th>Port B</th>
<th>MODx write</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Special Single chip</td>
<td>In/Out</td>
<td>In/Out</td>
<td>Write anytime, not peripheral</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Emulation Exp. narrow</td>
<td>A15-A8/ D7-D0</td>
<td>A7-A0</td>
<td>Cannot change</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Special test Exp. narrow</td>
<td>A15-A8/ D15-D8</td>
<td>A7-A0</td>
<td>Write anytime, not peripheral</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Emulation Exp. wide</td>
<td>A15-A8/ D15-D8</td>
<td>A7-A0</td>
<td>Cannot change</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Normal Single chip</td>
<td>In/Out</td>
<td>In/Out</td>
<td>Write once, Norm exp N/W</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>Normal Exp. narrow</td>
<td>A15-A8/ D7-D0</td>
<td>A7-A0</td>
<td>Cannot change</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Peripheral</td>
<td>–</td>
<td>–</td>
<td>Cannot change</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Normal Exp. wide</td>
<td>A15-A8/ D15-D8</td>
<td>A7-A0</td>
<td>Cannot change</td>
</tr>
</tbody>
</table>
MC9S12C32 Clock Circuit
MC9S12C32 Expanded Mode Bus Timing
Address Latch for MC9S12C32
General Approach to Memory Interfacing
Wide Expanded Mode
8K RAM Read Timing

- $A_{12}-A_0$
- $E_1$
- $G$
- $D_7-D_0$
- $Adv$
- $AdN$
- $t_{AVQV} = 150$
- $t_{E1LQV} = 150$
- $t_{GLQV} = 70$
- $t_{AXQX} = 20$
- $t_{E1HQZ} = 0$
- $t_{GHQZ} = 0$
- Read data available
8K RAM Write Timing

AdV

A12-A0

AdN

E1

W

D7-D0

\[ t_{DVWH} = 60 \]

\[ t_{WHDX} = 0 \]

Write data required
8K RAM Write Timing

\begin{align*}
\text{AdV} \quad & \quad \text{AdN} \\
\text{A12-A0} \quad & \quad \text{XXX} \\
\overline{W} \quad & \quad \text{XXX} \\
\overline{E1} \quad & \quad \text{XXX} \\
\text{D7-D0} \quad & \quad \text{XXX} \quad \text{Write data required}
\end{align*}

\[ t_{\text{DVWH}} = 60 \quad t_{\text{WHDX}} = 0 \]
8K RAM Interface ($8000-$9FFF)
8K RAM Interface Read Timing

Diagram showing the timing for an 8K RAM interface read operation, including signals E, R/W, and address lines A15-A8, A7-A0, with timing intervals for E1, RDA, and RDR.
8K by 16-bit RAM Interface
### Dynamic RAM (DRAM)

<table>
<thead>
<tr>
<th>DRAMs</th>
<th>SRAMs</th>
</tr>
</thead>
<tbody>
<tr>
<td>High density</td>
<td>Low density</td>
</tr>
<tr>
<td>One xtor, one cap./bit</td>
<td>3-4 xtors/bit</td>
</tr>
<tr>
<td>Slower</td>
<td>Faster</td>
</tr>
<tr>
<td>High fixed cost (refresh)</td>
<td>Low fixed cost (address decoder)</td>
</tr>
<tr>
<td>Low incremental cost</td>
<td>Higher incremental cost</td>
</tr>
<tr>
<td>Address multiplexing</td>
<td>Direct addressing</td>
</tr>
</tbody>
</table>

Scott R. Little (Lecture 22: Memory Interfacing)