Lecture 18:
Input and Output
Today’s Goals

• Input and Output (I/O)
  – Port-mapped I/O
  – Memory-mapped I/O

• How to use I/O of HCS12 on Dragon12+
What are Input and Output?

• I/O allows microprocessors to communicate with other devices such as switches, LCD screens, and keypads.

• Important terms
  – Pin and Port:
    – Directions
    – Full-duplex and Half-duplex
  – Asynchronous and Synchronous
Accessing I/O

Memory-mapped vs. Port-mapped I/O

• I/O can be accessed by a program just much like accessing memory addresses.

• There are two different approaches.
  – Memory-mapped I/O
  – Port-mapped I/O
I/O in HCS12

- **Ports**

- These ports can be used as general-purpose I/O.
Ports B, H, and P

General purpose (used either input or output)

- There are 8 pins in each of these ports (B, H, and P).

- Each port has a corresponding memory address that shows the values of the 8 pins.
  - B: $0001
  - H: $0260
  - P: $0258

- When they are used as input ports

- When they are used as output ports
  - The program stores a value into the location
  - The hardware sets the voltage according to the value.

- How do we determine if a port is being used as input or output?
Data Direction Registers

- Data direction registers
  - A general purpose I/O must be set whether input or output.

- Each port has its own register and the pins of each port can be configured separately.
  - $0003: DDR for B
  - $0262: DDR for H
  - $025A: DDR for P

- To configure the ports, store a value into the corresponding DDR based on the values below.

- Note:
  - When a pin is configured for an input, storing a value to its data bit is ignored.
  - When a pin is configured for an output, the voltage at the pin is ignored.
Port B, P, and H in the Dragon12+

Port B – 7 segment digits

- HCS12 has already been connected to hardware.
- Port B
  - Port B supplies the values to the 7-segment digits
  - Each digit actually has 8 LEDs including decimal point.
  - The diagram shows which bit controls each LED.
  - The pins of Port B are connected to all four digits in the Dragon12+.

How can we display different numbers on four 7-segments?
Port B, P, and H in the Dragon12+

Port P – Selecting a 7-segment digit

- 7 segment digit selection
  - Port P is used to select which of the four 7-segment LED digits are enabled.
  - Remember that the display pattern is determined by Port B.
  - Those digits that are not selected will be off (all LEDs off).
  - Note: only lower 4 bits are used.

- Enable/Disable
Port B, P, and H in the Dragon12+

Port H – switch input

- Port H is used to read the 8-DIP switches and 4 push buttons.
  - Only four pins that monitor both switches.
  - Note 1: No way to distinguish which is being pressed.

- DIP switches
  
  ![DIP Switches Image]

- Push Buttons
  
  ![Push Buttons Image]
Example Program

• Write a program
  – Turns on one LED segment of DIGIT3 at a time.
  – When the program begins, only segment 0 should be on.
  – Every time BUTT3 is pressed, the current LED segment turns off and the next one (see the numbers on the figure) turns on.
DATASTART  equ  $1000
PROGSTART  equ  $2000

; Ports on Dragon12+
PORTB   equ  $0001
DDRB    equ  $0003
PORTH   equ  $0260
DDRH    equ  $0262
PORTP   equ  $0258
DDRP    equ  $025A
Switch Bounce

• When a switch is asserted, we expect a signal something like the top right picture.

• However, signals has a transient period.

• When a switch (or button) is asserted (or pressed), the actual signal can be the bottom right figure.
  – For a short period of time, the switch signal is bouncing.

• That is why the program detects multiple buttons.
PORTB EQU $0001
DDRB EQU $0003
PORTH EQU $0260
DDRH EQU $0262
PORTP EQU $0258
DDRP EQU $025A

ORG $2000
### File: d12plus.inc

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Equate To</th>
<th>Hexadecimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>PORTB</td>
<td>EQU</td>
<td>$0001</td>
</tr>
<tr>
<td>DDRB</td>
<td>EQU</td>
<td>$0003</td>
</tr>
<tr>
<td>PORTH</td>
<td>EQU</td>
<td>$0260</td>
</tr>
<tr>
<td>DDRH</td>
<td>EQU</td>
<td>$0262</td>
</tr>
<tr>
<td>PORTP</td>
<td>EQU</td>
<td>$0258</td>
</tr>
<tr>
<td>DDRP</td>
<td>EQU</td>
<td>$025A</td>
</tr>
<tr>
<td>DOUT</td>
<td>EQU</td>
<td>$FF</td>
</tr>
<tr>
<td>DINP</td>
<td>EQU</td>
<td>$00</td>
</tr>
<tr>
<td>DIGIT0</td>
<td>EQU</td>
<td>$FE</td>
</tr>
<tr>
<td>DIGIT1</td>
<td>EQU</td>
<td>$FD</td>
</tr>
<tr>
<td>DIGIT2</td>
<td>EQU</td>
<td>$FB</td>
</tr>
<tr>
<td>DIGIT3</td>
<td>EQU</td>
<td>$F7</td>
</tr>
<tr>
<td>BUTT0</td>
<td>EQU</td>
<td>$08</td>
</tr>
<tr>
<td>BUTT1</td>
<td>EQU</td>
<td>$04</td>
</tr>
<tr>
<td>BUTT2</td>
<td>EQU</td>
<td>$02</td>
</tr>
<tr>
<td>BUTT3</td>
<td>EQU</td>
<td>$01</td>
</tr>
</tbody>
</table>

**INCLUDE d12plus.inc**

**ORG $2000**
;******************************************************************************
; Purpose:
; Define constants for D12PLUS
;
; History:
; 2/21/2010: Prof. Kwon created
;
;******************************************************************************

; Memory mapping of Dragon12+
DATASTART  equ  $1000
PROGSTART   equ  $2000

; Constants
TRUE       equ  $FF
FALSE      equ  $00

; Ports on Dragon12+
PORTB       equ  $0001
DDRB       equ  $0003
PORTH       equ  $0260
DDRH       equ  $0262
PORTP       equ  $0258
DDRP       equ  $025A

; Logical name of the ports on Dragon12+
SEGPATTN    equ  PORTB
PSHBUTTON    equ  PORTH
DIGITNUM     equ  PORTP

; General I/O port configuration
DOUT         equ  $FF
DINP         equ  $00
; DIGIT 7-segment LEDs on Dragon12+
; DIGIT3, DIGIT2, DIGIT1, DIGIT 0 from the left
DIGIT0 equ $F7 ; %11110111
DIGIT1 equ $FB ; %11111011
DIGIT2 equ $FD ; %11111101
DIGIT3 equ $FE ; %11111110

; Push buttons on Dragon12+
; BUTT3, BUTT2, BUTT1, BUTT0 from the left
BUTT0 equ $08 ; %00001000
BUTT1 equ $04 ; %00000100
BUTT2 equ $02 ; %00000010
BUTT3 equ $01 ; %00000001

File: lec18.asm

INCLUDE d12plus.inc

MXDLY equ $6000

ORG PROGSTART
Questions?
Wrap-up

What we’ve learned
What to Come