Lab 4:
Seven-segment LEDs

黃永廣
助教：劉柏廷、郭柏興、林嘉偉、陳昶宏
雲科電子

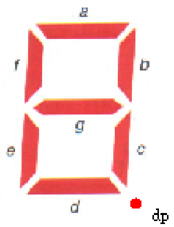
2014/12/1~5

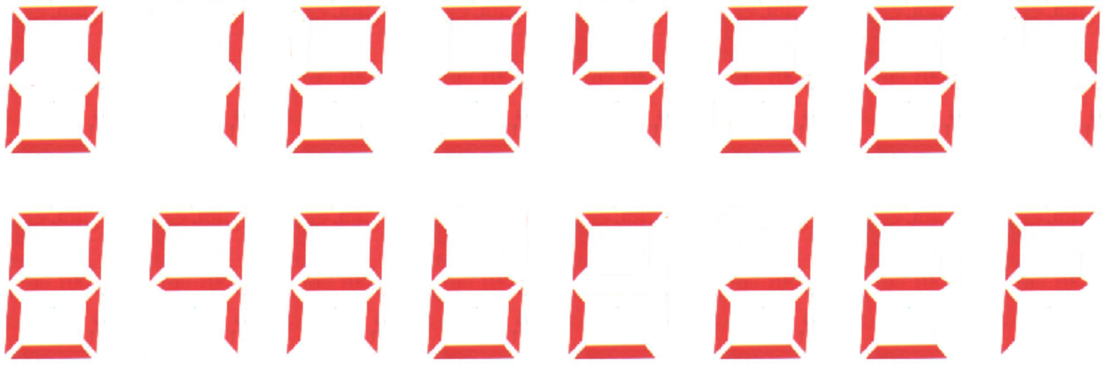
7-segments

- There are four 7-segments on DE0 FPGA board
- For this lab, the LC3 core does not do encoding for 7-segments
- 8-bits are used to control the 7 segments and one dot at the bottom right corner
- We need to use a table to look up the 8-bit code of a 4-bit number to control one 7-segment
- On DE0, we must control Hex10 with a 16-bit (two 8-bit codes) number and another 16-bit number for Hex32

共陽七段顯示器

規格：共陽七段顯示器





- FF_Tbl .FILL xFFC0 ; code for 0 is xC0 (1100 0000) ; only h (dp) and g are off
- hgfe dcba
- ; FF is for turning off upper half-word
- .FILL xFFF9 ; code for 1 is xF9 (1111 1001) ; only c and b are on
- … ; hgfe dcba

Show x1 on Hex10

- AND R1,R1,#0
- ADD R1,R1,#1
- LEA R7 FF_Tbl ; R7 := address of FF_Tbl
- ADD R3,R7,R1 ; R3 := address of ‘1’ = FF_Tbl+1
- LDR R2,R3,#0 ; R2 := m[R3] = code of ‘1’=xFFF9
- AND R0, R0, #0 ; R0 := #0
- STR R2, R0, #-5 ; Hex10 = m[xFFFB] := R2 = code of ‘1’
- …
- FF_Tbl .FILL xFFC0 ; code for 0 is xC0,
- ; FF is for turning off upper half-word
- .FILL xFFF9 ; code for 1 is xF9
- … ; code for 2, 3, 4,…

Show x10 on Hex10 and Hex32

- Assume R0 = 8-bit code with leading FF, e.g., xFFC0 (C0 is code for 0)
- Assume R1 = 8-bit code with trailing FF, e.g., xF9FF (F9 is code for 1)
- How to combine the two codes in order to show two digits on Hex1_Hex0?
- AND R2,R0,R1
  - E.g., R2 := (xFFC0 and xF9FF) = xF9C0
- AND R0, R0, #0
- STR R2, R0, #-5   ; Hex10 := R2
- STR R2, R0, #-6   ; Hex32 := R2

Java: Sum of the Elements of an Array

- int[] ray = new int[5];
- int sum = 0;
- for (int i=4; i>=0; i--)

      sum+= ray[i];

LC-3: Sum of the Elements of An Array (I)

```
int[] ray = new int[5];
int sum = 0;
for (int i=4; i>=0; i--)
   sum+= ray[i];
```

```
; lc-3 program loads at x3000
; array 'ray' starts at x3010
; R1 : address of ray
; R2 : i
; R3 : sum
;
```

```
     .ORIG x3000
     LEA R1,xF        ;[x3000]
     AND R2,R2,#0 ;[x3001]
     ADD R2,R2,#4 ;[x3002]
     AND R3,R3,#0 ;[x3003]
Loop ADD R4,R2,R1 ;[x3004]
     LDR R5,R4,#0  ;[x3005]
     ADD R3,R3,R5 ;[x3006]
     ADD R2,R2,-1  ;[x3007]
     BRzp Loop        ;[x3008]
…
     .END
```

LC-3: Sum of the Elements of An Array (II)

```
•      .ORIG x3000
•      LEA R1,xF        ;[x3000] ; initialize R1 to address of ray
•      AND R2,R2,#0 ;[x3001]
•      ADD R2,R2,#4 ;[x3002] ; i = 4
•      AND R3,R3,#0 ;[x3003] ; sum = 0
•   Loop ADD R4,R2,R1 ;[x3004] ; R4 = address of ray[i]
•      LDR R5,R4,#0  ;[x3005] ; R5 = ray[i]
•      ADD R3,R3,R5 ;[x3006] ; sum += ray[i]
•      ADD R2,R2,-1  ;[x3007] ; i--
•      BRzp Loop        ;[x3008] ; loop if i > 0
```

Task 1: Show x0 to xF on Hex10

- Construct a table of encodings for hexadecimal digits 0,1,2,…,F
- Construct one code table with xFFC0,…
- Show the numbers x0, x1, x2, x3,…, xE, xF on Hex10 and then repeats forever

Task 2: Show 4 Digits on Hex3210

- Construct a table of encodings for hexadecimal digits 0,1,2,…,F
- Hint: construct two tables, one with xFFC0,…, and another with xC0FF,…
- This is easier for combining two 8-bit codes into one word (16 bits)
- Define the last four digits of your student number in one memory location NUM
- Show the number in Hex32, Hex10

Task 2 Idea

- …
- LD R2, Num
- LD R3, Mask0
- AND R4,R2,R3 ; Get rightmost digit (1) of 0x4321
- … ; Get 2nd rightmost digit (2) of 0x4321
- ; Get 3rd rightmost digit (3) of 0x4321
- ; Get 4th rightmost digit (4) of 0x4321
- ; Show digits on Hex10 and Hex32
- Bye BRNZP Bye

- Num .FILL 0x4321 ; the last 4 digits of your student no.
- Mask0 .FILL 0x000F
- .END

Task 2 Idea

```
…
 LD R2, Num
 LD R3, Mask0
 AND R4,R2,R3 ; Get rightmost digit (1) of 0x4321
 …              ; Get code of 1 (0xFFF9) from FF_Tbl
 LD R3,Mask1
 AND R4,R2,R3 ; R4=x20;  counter =0
 …            ; x20-x10 = x10;  counter=counter+1=1
              ; x20-x10 = x00;  counter=counter+1= 2
              ; Get code of 2 (0xA4FF) from Tbl_FF
              ; 0xFFF9 AND 0xA4FF = 0xA4F9
              ; store 0xA4F9 into memory M[xFFFB] shows x21 on HEX10

Bye BRNZP Bye

Num .FILL 0x4321   ; the last 4 digits of your student no.
Mask0 .FILL 0x000F
Mask1 .FILL 0x00F0
 .END
```

Task 3: Countdown Counter

- Show x000A on Hex32, Hex10
- Each time, subtract one from the value and show on Hex32, Hex10
- When x0000 is shown at time t, xFFFF should be shown at time t+1
- When xFFF0 is shown at time t, xFFEF should be shown at time t+1
- When xFF00 is shown at time t, xFEFF should be shown at time t+1

Task 4:
Slow Decimal Countdown Counter

- Do Task 3 except the counter is decimal (instead of hexadecimal)
- Start the counter at 0010, then at 0009, 0008, …, 0000, 9999, 9998, …
- Do this in a slow way: do the counting as in Task 3 by computing the decimal of the hexadecimal counter and then display the four decimal digits.

Task 5:
Faster Decimal Countdown Counter

- Similar to Task 4 in display from 0010, 0009,…, 0000, 9999, 9998,…
- Store the four decimal digits in four memory locations
- Loop: If the unit digit is not zero, then subtract 1 from the unit digit
- Else borrow one from the next digit and repeat borrowing in the next digit if necessary
- Display the four decimal digits
- Goto Loop