Lab 2: Computing N^M

黃永廣
助教：劉柏廷、郭柏興、林嘉偉、陳昶宏
雲科電子
2014/10/20~10/24

Task 1: Reading N, M from Switches

- Use switches SW[5:3] to input a 3-bit number N
- Use switches SW[2:0] to input a 3-bit number M
- Show {b0000, SW[2:0], SW[5:3]} into LED[9:0]
- Repeat to execute the program from the first instruction

- ; R1 := SW[5:3], e.g., R1 := x7
- ST R1, N
- …
- Bye BRnzp Bye
- N .FILL #0

How to Get the Number of SW[2:0]?

- Mask := x0007 = b0000000000000111
- V := SW[7:0]
- M := V and Mask

How to Get the Number from SW[5:3]?

- Mask := b00 111 000
- Y := SW[7:0]
- Y := Y and Mask
- N := 0
- While (Y >= 8) {
- Y := Y-8
- N := N+1
- }

- For example, SW[7:0] = b11 111 101
- Y := SW[7:0] and Mask
- = 11 111 101 and 00 111 000 = 00 111 000
- = 32+16+8 = 56
- Y = 56 -> 48 -> 40 -> 32 -> 24 -> 16 -> 8 -> 0
- N = 0 -> 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7 So N = 7 = b111

- .ORIG x3000
- LD R1, Mask
- ; error if do
- ; AND R2,R2,x38
- …
- Bye BRnzp Bye
- Mask .FILL x38

How to Get the Number from SW[7:6]?

- Y := SW[7:0]
- N := 0
- While (Y >= 64) {
- Y := Y-64
- N := N+1
- }

- For example, SW[7:0] = b11 111 101
- Y := SW[7:0]
- = 11 111 101
- = 128+64+32+16+8+4+1=192+61=253
- Y = 253 -> 199 -> 125 -> 61
- N = 0 -> 1 -> 2 -> 3 So N = 3

How to Combine Bits?

- Given
- M := SW[2:0]
- N := SW[5:3]
- Want to get the number Y {SW[2:0], SW[5:3]}
- $Y := M*2^3 + N$
- E.g., SW[5:0] = b111 110; M := b110; N := b111
- $Y := M*2^3 + N = 6*8 + 7 = 53 =$ 110 111

How to Compute $M * 2^K$?

- Given M and K
- Ans := M
- While (K>0) {
- Ans := Ans + Ans
- K := K-1
- }

- M = 5; K = 3
- Ans := M = 5
- K=3 > 0
- Ans := Ans + Ans = 10
- K := K – 1 = 2 > 0
- Ans := Ans + Ans = 20
- K := K – 1 = 1 > 0
- Ans := Ans + Ans = 40
- K := K – 1 = 0
- Ans = 40 = $5 * 2^3$

Task 2: Compute N*M

- N := SW[7:4]
- M := SW[3:0]
- LED[9:0] := N * M

- E.g., SW[7:0] = 8'b0011_0101
- N := SW[7:4] = 4'b0011 = #3
- M := SW[3:0] = 4'b0101 = #5
- N*M = #3 * #5 = #15
- LED[9:0] = 10'b00_0000_1111

Task 3: Compute N^M

- N := SW[7:4]
- M := SW[3:0]
- X := M

- Ans := 1
- While (X > 0) {
- Ans := Ans*N
- X := X-1
- }
- LED[9:0] := Ans

- Example
- N := 3
- M := 4
- X := 4
- Ans := 1
- X=4 > 0
- Ans := 1*N = 3
- X := X-1=3 > 0
- Ans := 3*N = 9
- X := X-1=2 > 0
- Ans := 9*N = 27
- X := 2-1=1 > 0
- Ans := 27*N = 81
- X := 1-1=0 == 0
- LED[9:0] := Ans = 81

LED[9:0] := SW[3:0]

```
    .ORIG x3000                    ; file = "lab1a.asm"
Loop  LD R1, Saddr  ; R1 ←m[Saddr] = 0xFFFC = addr of SWs
    LDR R3, R1, #0      ; R3 ←m[0xFFFC+0] ; read switches
    AND R3,R3,x000F  ; R3 ←R3 and x000F
    LD R2, Laddr        ; R2 ←m[Laddr] = 0xFFFD = addr of LEDs
    STR R3, R2, #0      ; m[0xFFFD+0] ← R3 = switches; write LEDs
    BRnzp Loop          ; Repeat the above loop
Bye BRnzp Bye
Saddr .FILL 0xFFFC   ; Memory address for switches
Laddr .FILL 0xFFFD   ; Memory address for LEDs
    .END
```

Clear All Registers at First

- .ORIG x3000
- AND R0,R0,#0
- AND R1,R1,#0
- AND R2,R2,#0
- AND R3,R3,#0
- AND R4,R4,#0
- AND R5,R5,#0
- AND R6,R6,#0
- AND R7,R7,#0