EECE416 Microcomputer Fundamentals   Dr. Charles Kim

Assignment 2 (80 points)

A. Questions

(a). Find the on-chip memory size (in KB, MB, or GB)for the ARM chip with the following address range:

1. 0x00000000 – 0x00FFFFFF

   HA-LA + 1 = 01000000 → $16^6=(2^4)^6=2^{26}=2^4*2^{20}$ = 16 MB

2. 0x00100000 – 0X0027FFFF

   HA-LA+1 = 00180000 → $16^5+8*16^4 = 2^{20}+2^{19}=$ 1.5 MB

3. 0x02001000 – 0x03FFFFFF

   HA-LA+1 = 01FFF000 → $16^6 + 15*16^5 + 15* 16^4 + 15*16^3$ = (16+15+15/16+15/128) →32 MB

4. 0x4000F000 – 0x4004FFFF

   HA-LA+1 = 00041000 → $4*16^4 + 1*16^3= 2^{10}(2^8+2^2)=$260KB

(b). Write a simple code for each of the following instruction.

5. Add (a) the last 4 digits of your ID number and (b) the last 4 digits your ID number in reverse order as decimal, and store the sum in **r1**. Ignore the leading 0 in your ID number.

6. Store the last 4 digits of your ID to the memory location 0x20000010, and the first 4 digits of your ID to 0x20000014.

7. (from the result of 6 above) Load a word value to **r5** from the location of 0x20000010 and add it to your

entire ID as a decimal number. And store the sum to **r6**.

8. (continued from 7 above) Store the contents of **r6** to the memory location of 0x20001020.

Assign2.s

```
.text

.equ HUID, 3456789 //excluding the leading 0
.equ firstFour, 3456
.equ lastFour, 6789
.equ memLoc1, 0x20000010
.equ memLoc2, 0x20000014
.equ memLoc3, 0x20001020

.global _start
_start:
//#5 Add the following 2 decimal numbers: (a) the last 4 digits of your ID number
// and (b) the last 4 digits your ID number in reverse order.
// Ignore the leading 0 in your ID number.
// And then store the sum in r1.
	LDR r2, =firstFour
	LDR r3, =lastFour
	ADD r1, r2, r3
// 6. Store the last 4 digits of your ID to the memory location 0x20000010,
// and the first 4 digits of your ID to 0x20000014.
	LDR r0,=memLoc1
	STR r3, [r0]  //lastFour
	LDR r0, =memLoc2
	STR r2, [r0] //firstFour
//7. (from the result of 6 above) Load a word value to r5 from the location of
0x20000010
// and add it to your entire ID as a decimal number.  And store the sum to r6

	LDR r0,=memLoc1
	LDR r5, [r0]
	LDR r6,=HUID
	ADD r6, r5

//8. (continued from 7 above) Store the contents of r6 to the memory location of
0x20001020.
	LDR r0, =memLoc3
	STR r6, [r0]


stop:
	B stop
```

```
.end
```

B. Score Distribution and Scoring Rubric: Total points = 80

	#1 - #8
10 pts	Correct with all calculation (or Assembly code) neatly displayed
6 pts	Incorrect (partially correct) with all calculations (or Assembly code) neatly displayed
4 pts	Correct without calculation (or Assembly coding)
0 pts	Incorrect without calculation (Assembly coding)

C. Submission: Submit 2 files. Part(a) for #1 - #4: Work on paper and submit your work by bringing it to the class or my office. Submission of a scanned copy of the manual work via email is also accepted. In the latter option, name the scanned copy as **416Assign2_LastName.xxx** (xxx being file type such as docx, doc, pdf, png, etc.). Part (b) for Assembly Code: Write the codes in the CPUlator emulator and store into one **s** file, **416Assign2_LastNamme.s**. Submit the above 2 files via email.

D. Submission due: 5:00pm (F) September 29, 2023 (Check the webpage)

E. Point Deduction on Late Submission (or **Maximum score by submission time**)

Submission Time/Date	Maximum score
By 5:00pm submission date	80
By 5:00pm submission date + 3	50
By 5:00pm submission date + 4	30
By 5:00pm submission date + 5	20
After the above	0