CISC 260 Machine Organization and Assembly Language (Spring 2019)

Assignment # 6

(Due: May 14, 2019)

Problem 1 [20 points]:

b.	Procedure A				Procedure B			
	Text Segment	Address	Instruction		Text Segment	Address	Instruction	
		0	LDR r0, [r3,#0]			0	STR r0, [r3,#0]	
		4	ORR r1, r0, #0			4	B 0	
		8	BL 0			…	…	
		…	…			0x180	MOV pc, lr	
						…	…	
	Data Segment	0	(X)		Data Segment	0	(Y)	
		…	…			…	…	
	Relocation Info	Address	Instruction Type	Dependency	Relocation Info	Address	Instruction Type	Dependency
		0	LDR	X		0	STR	Y
		4	ORR	X		4	B	FOO
		8	BL	B				
	Symbol Table	Address	Symbol		Symbol Table	Address	Symbol	
		–	X			–	Y	
		–	B			0x180	FOO	

Assume that Procedure A has a text size of 0x140, data size of 0x40 and Procedure B has a text size of 0x300 and data size of 0x50. Also assume static base pointer (SB) r9 = 0x1000 0000, and the memory allocation strategy as shown in Figure 6.30 in the textbook. Link the object files above to form the executable file, by merging A with B (A goes first). Give explicitly the addresses for instructions and data in the executable and update the symbol table.

Problem 2. [20 points] Assume for a given processor the CPI of arithmetic instructions is 1, the CPI of load/store instructions is 10, and the CPI of branch instructions is 3. Assume a program has the following instruction breakdowns: 500 million arithmetic instructions 300 million load/store instructions, 100 million branch instructions.

a) Suppose that new, more powerful arithmetic instructions are added to the instruction set. On average, through the use of these more powerful arithmetic instructions, we can reduce the number of arithmetic instructions needed to execute a program by 25%, while increasing the clock cycle time by only 10%. Is this a good design choice? Why?

b) Suppose that we find a way to double the performance of arithmetic instructions. What is the overall speedup of our machine? What if we find a way to improve the performance of arithmetic instructions by 10 times?

Problem 3. [20 points] Assume that for a given program 70% of the executed instructions are arithmetic, 10% are load/store, and 20% are branch.

a) Given this instruction mix and the assumption that an arithmetic instruction requires two cycles, a load/store instruction takes six cycles, and a branch instruction takes three cycles, find the average CPI.

b) For a 25% improvement in performance, how many cycles, on average, may an arithmetic instruction take if load/store and branch instructions are not improved at all?

c) For a 50% improvement in performance, how many cycles, on average, may an arithmetic instruction take if load/store and branch instructions are not improved at all?

Problem 4. [40 points]

a) [15 points] Modify the following C code so that the recursion is tail-recursion.

b) [25 points] Translate the tail recursion version into ARM assembly code.

```
funct(int x) {

  if (x <= 0) return 0;
  else if (x & 0x1) {
     return x + funct(x-1);
  } else {
     return x - funct(x-1);
  }
}
```

Bonus Problem [30 pts]. Write in ARM7 assembly language a program that takes as input a string, check if it is a palindrome or not, and print your answer as “Palindrome” or “Not palindrome” correspondingly.