The University of Alabama in Huntsville
Electrical and Computer Engineering Department
CPE 221 01
Sample Test 2

This test is closed book, closed notes. You may not use a calculator. You should have the 6 page ARM Instruction Reference. You must show your work to receive full credit.

Name: ___________________________

1. (1 point) A _________________________ points to a place on the stack and does not change throughout the lifetime of an instance of a procedure/
2. (1 point) A __________________________ control unit runs a program whose input is the machine-level op-code to be executed and whose output is the bus enables, multiplexer controls, clocks, and signal that control the processor.
3. (1 point) _________________ is a technique in which the execution of multiple instructions are overlapped to increase the number of instructions executed in a period of time.
4. (1 point) ___________________ are events that force the computer to stop normal processing and to invoke the operating system.
5. (1 point) ___________ is a load/store RISC ISA that has 32 general purpose registers.
6. (20 points) Write the code to implement the expression A = ((B/F) + (C × D)) × E on 3-, 2-, 1-, and 0-address machines. Do not rearrange the expression. In accordance with programming language practice, computing the expression should not change the values of its operands. When using a 0-address machine, the order used is SOS op TOS, where SOS is second on stack and TOS is top of stack.

7. (30 points) Consider the following ARM program. Trace the values of the registers shown as they change during program execution. Also, trace the writes to memory by the `STR` instruction and the stack activity. Clearly indicate the value of the sp. There may be unused columns or rows in the tables. If you need to add columns or rows, you may do so.

```
;   int main() {
;     int P = 3;
;     int Q = -1;
;     P = Func1(P);
;     Q = Func1(Q);}
;   int Func1(int x) {
;     if (x > 0) x = Times16(x) + 1;
;     else x = 32*x;
;     return(x);}
;   int Times16(int x) {
;     x = 16*x;
;     return(x);}

        AREA NESTED_SUBROUTINE_STACK, CODE, READWRITE
        ENTRY
        ADR     r4, P
        ADR     r5, Q
        MOV     sp, #0
        MOV     fp, #0x0000C000
        LDR     r0, [r4]
        BL      Func1
        STR     r1, [r4]
        LDR     r0, [r5]
        BL      Func1
        STR     r1, [r5]
done    B       done
Func1   PUSH    {fp}
        CMP     r0, #0
        PUSHGT  {lr}
        PUSHGT  {r0}
        BLGT    Times16
        POPGT   {r1}
        POPGT   {lr}
        ADDGT   r1, r1, #1
        MOVLE   r1, r0, LSL #5
        POP     {fp}
        MOV     pc, lr
Times16 POP     {r7}
        MOV     r7, r7, LSL #4
        PUSH    {r7}
        MOV     pc, lr
        AREA  NESTED_SUBROUTINE_STACK, DATA, READWRITE
P       DCD   3
Q       DCD   -1
        END
```

r0													
r1													
r4													
r5													
r7													
lr													
fp													

Results of the `STR` instructions.

Memory Address	Contents

Address	Value
FFFF FFEC	
FFFF FFF0	
FFFF FFF4	
FFFF FFF8	
FFFF FFFC	

Instruction:

____

Address	Value
FFFF FFEC	
FFFF FFF0	
FFFF FFF4	
FFFF FFF8	
FFFF FFFC	

Instruction:

____

Address	Value
FFFF FFEC	
FFFF FFF0	
FFFF FFF4	
FFFF FFF8	
FFFF FFFC	

Instruction:

____

Address	Value
FFFF FFEC	
FFFF FFF0	
FFFF FFF4	
FFFF FFF8	
FFFF FFFC	

Instruction:

____

Address	Value
FFFF FFEC	
FFFF FFF0	
FFFF FFF4	
FFFF FFF8	
FFFF FFFC	

Instruction:

____

Address	Value
FFFF FFEC	
FFFF FFF0	
FFFF FFF4	
FFFF FFF8	
FFFF FFFC	

Instruction:

____

Address	Value
FFFF FFEC	
FFFF FFF0	
FFFF FFF4	
FFFF FFF8	
FFFF FFFC	

Instruction:

____

Address	Value
FFFF FFEC	
FFFF FFF0	
FFFF FFF4	
FFFF FFF8	
FFFF FFFC	

Instruction:

____

Address	Value
FFFF FFEC	
FFFF FFF0	
FFFF FFF4	
FFFF FFF8	
FFFF FFFC	

Instruction:

____

Address	Value
FFFF FFEC	
FFFF FFF0	
FFFF FFF4	
FFFF FFF8	
FFFF FFFC	

Instruction:

____

Address	Value
FFFF FFEC	
FFFF FFF0	
FFFF FFF4	
FFFF FFF8	
FFFF FFFC	

Instruction:

____

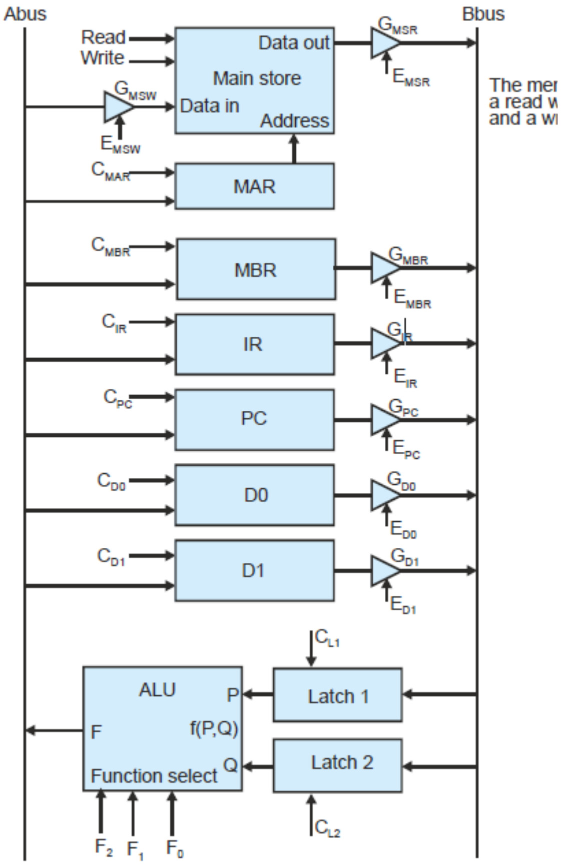
Address	Value
FFFF FFEC	
FFFF FFF0	
FFFF FFF4	
FFFF FFF8	
FFFF FFFC	

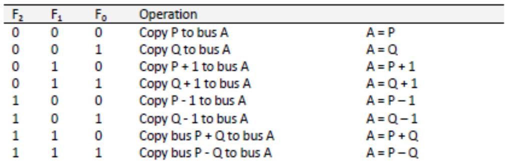
Instruction:

____

8. (20 points) For the architecture shown, write the sequence of signals and control actions necessary to execute the instruction STR D0, [P, D1], that adds the contents of the memory location pointed at by the contents of memory location P plus the contents of register D1 and uses that as the effective address to store the contents of D0.

M[M[P] + D1] ← D0





F_2	F_1	F_0	Operation	
0	0	0	Copy P to bus A	A = P
0	0	1	Copy Q to bus A	A = Q
0	1	0	Copy P + 1 to bus A	A = P + 1
0	1	1	Copy Q + 1 to bus A	A = Q + 1
1	0	0	Copy P - 1 to bus A	A = P – 1
1	0	1	Copy Q - 1 to bus A	A = Q – 1
1	1	0	Copy bus P + Q to bus A	A = P + Q
1	1	1	Copy bus P - Q to bus A	A = P – Q

Cycle	Concrete RTL	Signals
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		

9. (10 points) A processor executes an instruction in the following six stages. The time required by each stage in picoseconds (1,000 ps = 1 ns) is given for each stage.

IF	Instruction fetch	320 ps
ID	Instruction decode	200 ps
OF	Operand fetch	280 ps
OE	Execute	350 ps
M	Memory access	500 ps
OS	Operand store (writeback)	220 ps

a. What is the time to execute an instruction if the processor is not pipelined?

b. What is the time taken to fully execute an instruction assuming that this structure is pipelined in six stages and that there is an additional 15 ps per stage due to the pipeline latches?

c. Once the pipeline is full, what is the average instruction execution time?

d. Suppose that 30% of instructions are branch instructions that are taken and cause a 4-cycle penalty, what is the effective instruction execute time?

10. (15 points) A RISC processor executes the following code. There are data dependencies but no internal forwarding. A source operand cannot be used until it has been written.

```
ADD  r0, r1, r2
ADD  r3, r6, r4
ADD  r5, r3, r0
ADD  r7, r0, r8
ADD  r9, r6, r9
ADD  r0, r3, r5
```

a. Assuming a four-stage pipeline (fetch, operand fetch, execute, and result write), what registers are being read during the eighth clock cycle and what register is being written?

b. How long will it take to execute the entire sequence?