ELE 818 * ADVANCED COMPUTER ARCHITECTURES * MIDTERM TEST *

SAMPLE

1 Section: *Simple pipeline for integer operations*

For all following questions we assume that:
a) Pipeline contains 5 stages: IF, ID, EX, M and W;
b) Each stage requires one clock cycle;
c) All memory references hit in cache;
d) Following program segment should be processed:

```
// ADD TWO INTEGER ARRAYS
        LW      R4      # 400
L1:     LW      R1,     0 (R4)      ; Load first operand
        LW      R2, 400 (R4)        ; Load second operand
        ADDI    R3, R1, R2          ; Add operands
        SW      R3,     0 (R4)      ; Store result
        SUB     R4, R4, #4          ; Calculate address of next element
        BNEZ    R4, L1              ; Loop if (R4) != 0
```

Question # 1.1

Calculate how many clock cycles will take execution of this segment on the regular (non-pipelined) architecture. Show calculations:

Number of cycles = (2)

Question # 1.2

Calculate how many clock cycles will take execution of this segment on the simple pipeline without forwarding or bypassing when result of branch instruction (new PC content) is available after WB stage. Show timing of one loop turn on Fig 1.1:

Instruction	**Clock cycle number**																
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
LW R1, 0 (R4)																	
LW R2, 400 (R4)																	
ADDI R3,R1,R2																	
SW R3, 0 (R4)																	
SUB R4, R4, #4																	
BNEZ R4, L1																	

Fig 1.1

Number of cycles = (2)

Question # 1.3

Calculate how many clock cycles will take execution of this segment on the simple pipeline with normal forwarding and bypassing when result of branch instruction (new PC content) is available after completion of the ID stage. Show timing of one loop turn on Fig 1.2:

Instruction	**Clock cycle number**																
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
LW R1, 0 (R4)																	
LW R2, 400 (R4)																	
ADDI R3,R1,R2																	
SW R3, 0 (R4)																	
SUB R4, R4, #4																	
BNEZ R4, L1																	

Fig 1.2

Number of cycles = (2)

Question # 1.4

Schedule the segment instructions including branch-delay slot to get minimum processing time assuming that pipeline has normal forwarding and bypassing hardware. It is possible to reorder instructions and change position of loop label (L1) but not name of registers or opcode modification.

Show scheduled segment, position of L1 and pipeline timing diagram on Fig 1.3 and calculate number of clock cycles needed to execute this task segment.

Instruction	**Clock cycle number**																
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**

Fig 1.3

Number of cycles = (4)

Question # 1.5

Schedule the program segment presented on Fig. 1.4, including branch-delay slot to get minimum processing time assuming that pipeline has normal forwarding and bypassing hardware. It is possible to reorder instructions and change position of loop label (PR) but not name of registers or opcode modification.
Note: In the following segment branch is mostly taken.

```
1.      LW    R4,   0(R1)
2.      LW    R3, 20(R1)
3.      ADD   R2, R3, R4
4.      SW    R2,   0(R1)
5.      SUBI  R1,  R1, # 4
6.      BNEZ R1,  PR
7.      XOR   R8, R3, R4
8.      ADDI R8, R8, # 1000
9.  PR: ADDI  R6, R3, #10
10.     ADD   R6, R6, R2
11.     SW    R6,   0(R8)
```

Fig. 1.4

```
1._____
2._____
3._____
4._____
5._____
6._____
7._____
8._____
9._____
10._____
11._____
```

Fig. 1.5

a) Show on Fig. 1.4 using symbol - "→" positions of delay-slots (between instructions) if original segment will be processed on the simple integer pipeline with forwarding hardware and new PC content available after ID stage of any branch instruction. (2)

b) Show scheduled segment with new position of 'PR' on Fig 1.5 . (4)

2 **Section:** *Floating-point pipeline*

For all following questions we assume that:
a) Pipeline contains stages: IF, ID, EX, M and W;
b) Each stage except EX requires one clock cycle;
c) System contains 3 ALUs for Integer operations, FP-addition and FP-multiplication:
   EX-stage for Integer operations contains 1 clock cycle (EX);
   EX-stage for ADDD operation contains 2 clock cycles (A1,A2);
   EX-stage for MULTD operation contains 4 clock cycles (M1, M2, M3, M4);
d) All memory references hit in cache;
e) System has data and control hazard preventing subsystem.

Following task segment should be processed:

// Formula calculation : Y (I) = K*X (I) + B

```
1.              LD      F3,   0(R1)      ; Load value of 'K'
2.              LD      F4,   8(R1)      ; Load value of 'B'
3.      Loop:   LD      F2, 100(R0)      ; Load value of 'X (I)
4.              MULTD   F1, F2, F3       ; Calc. Value of 'K*X(I)'
5.              ADDD    F1, F1, F4       ; Calc. Value of Y(I)
6.              SD      F1, 200(R0)      ; Store Y(I)
7.              SUBI    R0, R0, # 8
8.              BNEZ    R0, Loop
```

Question # 2.1

Calculate how many clock cycles will take loop execution (from IF to IF of LD F2, ...) of this segment on the this FP-pipeline with normal forwarding and bypassing when result of branch instruction (new PC content) is available after completion of the ID stage. Show timing of the first loop turn (starting from IF of LD F3, 0(R1)) on Fig 2.1:

Instruction	**Clock cycle number** **1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18**
LD F3, 0(R1)	
LD F4, 8(R1)	
LD F2, 100(R0)	
MULTD F1,F2,F3	
ADDD F1,F1,F4	
SD F1, 200(R0)	
SUBI R0, R0, # 8	
BNEZ R0, Loop	
LD F2, 100(R0)	

Fig 2.1

Number of clock cycles in the loop = _____ (4);

Question # 2.2

Unroll above task segment using register renaming technique. Show unrolled segment filling the listing presented on Fig. 2.2 (See next page) (4);

Inst. #	**Instruction**
1	LD F3, 0(R1)
2	LD F4, 8(R1)
3	Loop : LD F2, 100(R0)
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	

Fig. 2.2

3 **Section:** *Hazards*

For all following questions we assume that:

a) Pipeline contains stages: IF, IS (Issue), RO (Read operand), EX and W(Write);
b) Each stage except EX requires one clock cycle;
c) System contains 4 FUs for FP operations, FP-load / store, FP-addition / subtraction FP-multiplication and FP-division:
   EX-stage for Load / Store operations contains 1 clock cycle (EX);
   EX-stage for ADDD or SUBD operations contains 1 clock cycle (A or S);
   EX-stage for MULTD operation contains 3 clock cycles (M1, M2, M3);
   EX-stage for DIVD operation contains 4 clock cycles (D1, D2, D3, D4);
d) All memory references hit in cache;
e) Pipeline has forwarding hardware for all FUs, except FP-Load / Store where operand is ready after W-stage;

Timing diagram of task segment processing is presented on Fig 3.1

Instruction	**Clock cycle number**																
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
LD F6, 20(R5)	IF	IS	RO	EX	W												
LD F2, 28(R5)		IF	IS	RO	EX	W											
MULTD F0,F2,F4			IF	IS	*	*	RO	M1	M2	M3	W						
SUBD F8,F6, F3				IF	IS	RO	S	W									
DIVD F10,F0,F6					IF	IS	*	*	*	*	R0	D1	D2	D3	D4	W	
ADDD F6, F8,F2						IF	IS	R0	A	W							
SD F8, 50(R5)							IF	IS	RO	EX	W						

Fig. 3.1

Question # 3.1

What kind of hazards there is between following instructions (Circle one of three):

1. LD F2, 28(R5) and MULTD F0,F2,F4: a) Structural, b) Data, c) Control. - (1)

2. DIVD F10,F0,F6 and ADDD F6, F8,F2: a) Structural, b) Data, c) Control. - (1)

3. MULTD F0,F2,F4 and SD F8, 50(R5) : a) Structural, b) Data, c) Control. - (1)

Question # 3.2

What kind of Data hazards there is between following instructions (Circle one of four):

1.MULTD F0,F2,F4 and DIVD F10,F0,F6: a) RAW, b) WAR, c) WAW, d) None – (1)

2. DIVD F10,F0,F6 and ADDD F6, F8,F2: a) RAW, b) WAR, c) WAW, d) None – (1)

3. MULTD F0,F2,F4 and SD F8, 50(R5) : a) RAW, b) WAR, c) WAW, d) None – (1)