Solutions for the Sample of Midterm Test

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:

Solution

Number of cycles = [Initial instruction + (Number of instructions in the loop L1) x number of loop cycles] x number of clock cycles / instruction (CPI) =

$$= [\ 1 + (\ 6\ ) \times 400/4\ ] \times 5 \text{ c.c.} = 3005 \text{ c.c.}$$

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 the branch instruction (new PC content) is available after WB stage. Show timing of one loop cycle in Figure 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																	

Figure 1.1

Solution

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

Comments:

1. Two stall cycles (c.c. # 5 and 6) are caused by the delay of data in the register R2 for the ADDI
2. Same stall cycles in ID stage for the SW instruction are because ID stage circuits are busy for ADDI and becoming available only on 7-th c.c.
3. SUB can start only on 8-th c.c. because IF stage is busy with SW instruction.
4. One c.c. stall in the pipeline happens because the content of R3 (for SW) is not ready. However, "Ex" stage can be executed for SW instruction. This becomes possible because during the "Ex" stage the address in memory is calculated (only for Load or Store instructions).
5. Two stall cycles (c.c. # 11 and 12) in BNEZ are coming from the delay of updating the R4. New content of R4 becomes available only after 12 c.c. Thus, the content of PC is updated on W-stage of BNEZ (after15 c.c.).

Number of cycles in the loop = 15 c.c.

Number of clock cycles for segment execution on pipelined processor =

= 1 c.c. (IF stage of the initial instruction) + (Number of clock cycles in the loop L1) x Number of loop cycles = 1 + 15 x 400/4 = 1501 c.c.

Speedup of the pipelined processor comparing with non-pipelined processor =

= Number of Clock cycles for the segment execution on non-pipelined processor / Number of Clock cycles for the segment execution on simple pipelined processor =

= 3005 c.c. / 1501 = 2 times

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 cycle in Figure 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																	

Figure 1.2

Solution

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

Comments:

1. Data (R2) for the ADDI is ready after "M" stage of the LW R2. During the "WB" stage the requested operand will be written to the R2 and operation register (e.g. Reg. A) of the ALU.
2. ID stage for the SW is delayed because it is busy with ADDI.
3. BNEZ can initiate IF stage of the LW R1, 0(R4) because new PC-content is ready after 8 c.c.

Number of cycles in the loop = 8 c.c.

Speedup of the pipelined processor with forwarding comparing with non-pipelined processor = 3005 c.c. / (1 c.c. + 400/4 x 8 c.c.) = 3005 / 801 = 3.75 times

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 op-code modification.
Show scheduled segment, position of L1 and pipeline timing diagram in Figure 1.3 and calculate number of clock cycles needed to execute this task segment.

Solution

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

Comments:
There are two time slots not used (stalls): during the 5-th and 8-th clock cycles.
Thus, if we reschedule the SUB instruction after the LW R2 then we will fill the 5-th c.c. slot. SW instruction may be moved to the end of the loop (after BNEZ) to fill the 8-th c.c. time slot. Now, there will not be any stalls and the minimum number of clock. cycles for the segment processing will be = 6 c.c.

The maximum speedup comparing with non-pipelined processor is =
= 3005 / (1+ 6 x 100) = 5 times
It means that all stages of 5-stage pipeline are always busy (no stalls) during the task segment execution.

---

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)) in Figure 2.1:

Instruction	**Clock cycle number**															
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
LD F3, 0(R1)	IF	ID	Ex	M	W											
LD F4, 8(R1)		IF	ID	Ex	M	W										
LD F2, 100(R0)			**IF**	ID	Ex	M	W									
MULTD F1,F2,F3				IF	ID	*	M1	M2	M3	M4	M	W				
ADDD F1,F1,F4					IF	*	ID	*	*	*	A1	A2	M	W		
SD F1, 200(R0)							IF	*	*	*	ID	Ex	*	M	W	
SUBI R0, R0, # 8											IF	ID	*	Ex	M	W
BNEZ R0, Loop												IF	*	ID	Ex	M
LD F2, 100(R0)															**IF**	ID

Figure 2.1 **I←** Period of one loop cycle **→I**

Number of clock cycles in the loop = 12 c.c.

Comments:

1. Stall cycle (c.c. # 6) is caused by the delay of data in the register F2 for the MULTD instruction
2. Same stall cycles in ID stage for the ADDD at c.c. # 5 is because ID stage circuits are busy for MULTD instruction and becomes available on 7-th c.c.
3. Three stall cycles (c.c. # 8,9 and 10) are caused by the delay of operand in F1 (result of MULTD instruction). It becomes available on c.c. # 11. Same reason delays ID stage for SD instruction
4. SUBI can start only on 11-th c.c. because IF stage is busy with SD instruction.
5. One c.c. stall in the pipeline happens at 13-th c.c. because of structural hazard (same M stage for ADDD and SD).

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 Figure 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						

Figure 3.1

Question # 3.1
What kind of hazards are between following instructions (Circle one of three):

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

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

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

Question # 3.2
What kind of Data hazards are between the following instructions (Circle one of four):

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

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

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