Cairo University **Faculty of Engineering**
Computer Engineering Department **CMP (N) 301 – Computer Architecture**

Sheet (3)
Chapter 4-Control Hazard

1) [4.10] In this exercise, we examine how resource hazards, control hazards, and ISA design can affect pipelined execution. Problems in this exercise refer to the following frag- ment of MIPS code:

```
SW R16,12(R6)
LW R16,8(R6)
BEQ R5,R4,Label ; Assume R5 != R4
ADD R5,R1,R4
SLT R5,R15,R4
```

assume that individual pipeline stages have the following latencies:

IF	ID	EX	MEM	WB
200ps	120ps	150ps	190ps	100ps

1) For this problem, assume that all branches are perfectly predicted (this eliminates all control hazards) and that no delay slots are used. If we only have one memory (for both instructions and data), there is a structural hazard every time we need to fetch an instruction in the same cycle in which another instruction accesses data. To guarantee forward progress, this hazard must always be resolved in favor of the instruction that accesses data. What is the total execution time of this instruction sequence in the 5-stage pipeline that only has one memory?
We have seen that data hazards can be eliminated by adding NOPs to the code. Can you do the same with this structural hazard? Why?

2) For this problem, assume that all branches are perfectly predicted (this eliminates all control hazards) and that no delay slots are used. If we change load/store instructions to use a register (without an offset) as the address, these instructions no longer need to use the ALU. As a result, MEM and EX stages can be overlapped and the pipeline has only 4 stages. Change this code to accommodate this changed ISA. Assuming this change does not affect clock cycle time, what speedup is achieved in this instruction sequence?

3) Assuming stall-on-branch and no delay slots, what speedup is achieved on this code if branch outcomes are determined in the ID stage, relative to the execution where branch outcomes are determined in the EX stage?

4) Given these pipeline stage latencies, repeat the speedup calculation from 2), but take into account the (possible) change in clock cycle time. When EX and MEM are done in a single stage, most of their work can be done in parallel. As a result, the resulting EX/MEM stage has a latency that is the larger of the original two, plus 20ps needed for the work that could not be done in parallel.

✓ 5) Given these pipeline stage latencies, repeat the speedup calculation from 3), taking into account the (possible) change in clock cycle time. Assume that the latency ID stage increases by 50% and the latency of the EX stage decreases by 10ps when branch outcome resolution is moved from EX to ID.

✓ 6) Assuming stall-on-branch and no delay slots, what is the new clock cycle time and execution time of this instruction sequence if BEQ address computation is moved to the MEM stage? What is the speedup from this change?
Assume that the latency of the EX stage is reduced by 20ps and the latency of the MEM stage is unchanged when branch outcome resolution is moved from EX to MEM.

2) [4.14] This exercise is intended to help you understand the relationship between delay slots, control hazards, and branch execution in a pipelined processor. In this exercise, we assume that the following MIPS code is executed on a pipelined processor with a 5-stage pipeline, full forwarding, and a predict-taken branch predictor:

```
lw r2,0(r1)
label1: beq r2,r0,label2 # not taken once, then taken
lw r3,0(r2)
add r2,r5,r6
beq r3,r0,label1 # taken
add r1,r3,r1
label2: sw r1,0(r2)
```

mas2la helwa awy bgd

✓ 1) Draw the pipeline execution diagram for this code, assuming there are no delay slots and that branches execute in the EX stage.

✓ 2) Repeat 1), but assume that delay slots are used. In the given code, the instruction that follows the branch is now the delay slot instruction for that branch.

✓ 3) Using the first branch instruction in the given code as an example, describe the hazard detection logic needed to support branch execution in the ID stage. Which type of hazard is this new logic supposed to detect? data hazards

✓ 4) For the given code, what is the speedup achieved by moving branch execution into the ID stage? Explain your answer. In your speedup calculation, assume that the additional comparison in the ID stage does not affect clock cycle time.