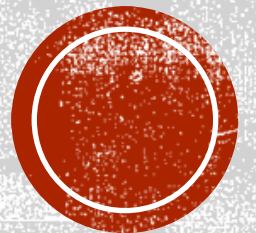


COMPUTER ORGANIZATION AND ARCHITECTURE

Course Code : CSE 2151

Credits : 04



MEMORY DELAYS

- Delays arising from memory accesses are another cause of pipeline stalls.
- Load instruction may require more than one clock cycle to obtain its operand from memory.
- This may occur because the requested instruction or data are not found in the cache, resulting in a cache miss.
- A memory access may take ten or more cycles. For simplicity, the figure shows only three cycles.

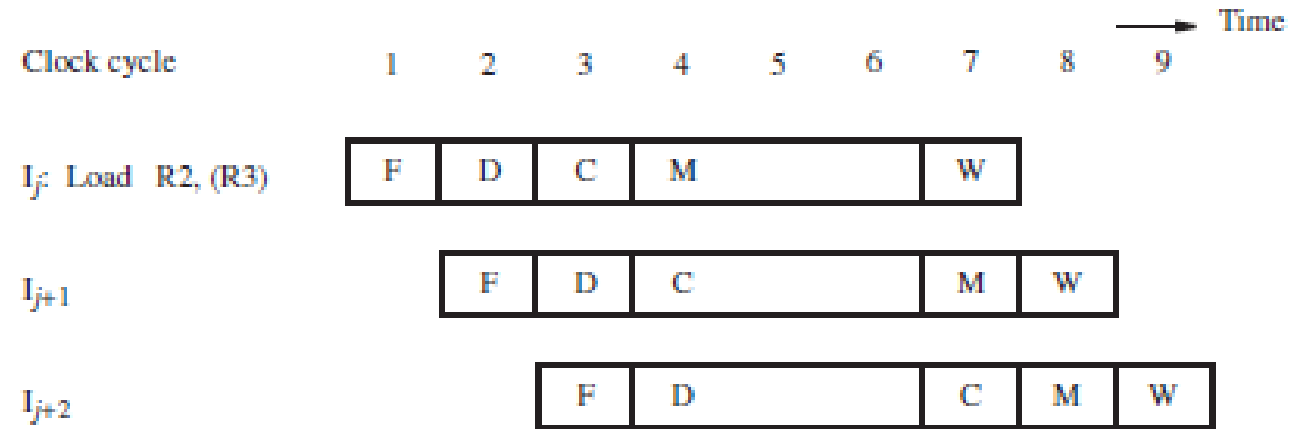


Figure 6.7 Stall caused by a memory access delay for a Load instruction.

MEMORY DELAYS

- Consider the instructions:
Load R2, (R3)
Subtract R9, R2, #30
- Assume that the data for the Load instruction is found in the cache, requiring only one cycle to access the operand.
- The destination register R2 for the Load instruction is a source register for the Subtract instruction.
- Operand forwarding cannot be done because the data read from memory (the cache, in this case) are not available until they are loaded into register
- the Subtract instruction must be stalled for one cycle

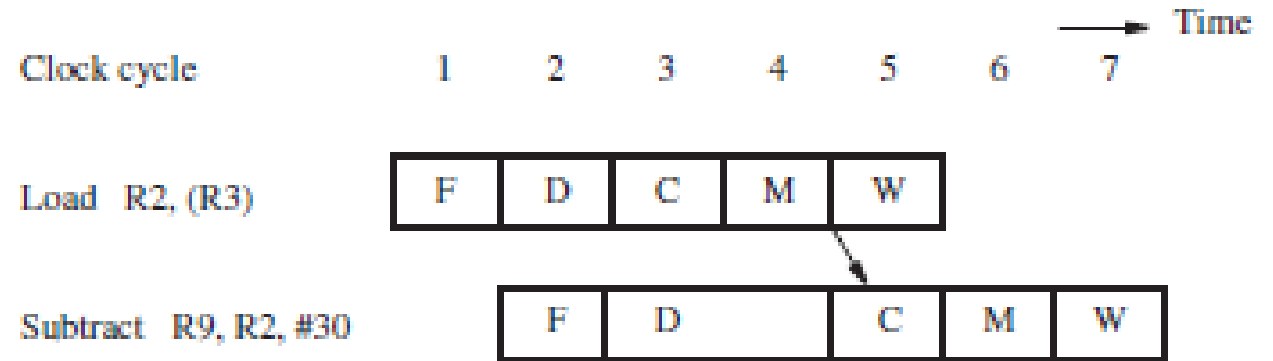


Figure 6.8 Stall needed to enable forwarding for an instruction that follows a Load instruction.

INSTRUCTION HAZARDS: OVERVIEW

- Whenever the stream of instructions supplied by the instruction fetch unit is interrupted, the pipeline stalls.
 - Branch

UNCONDITIONAL BRANCH

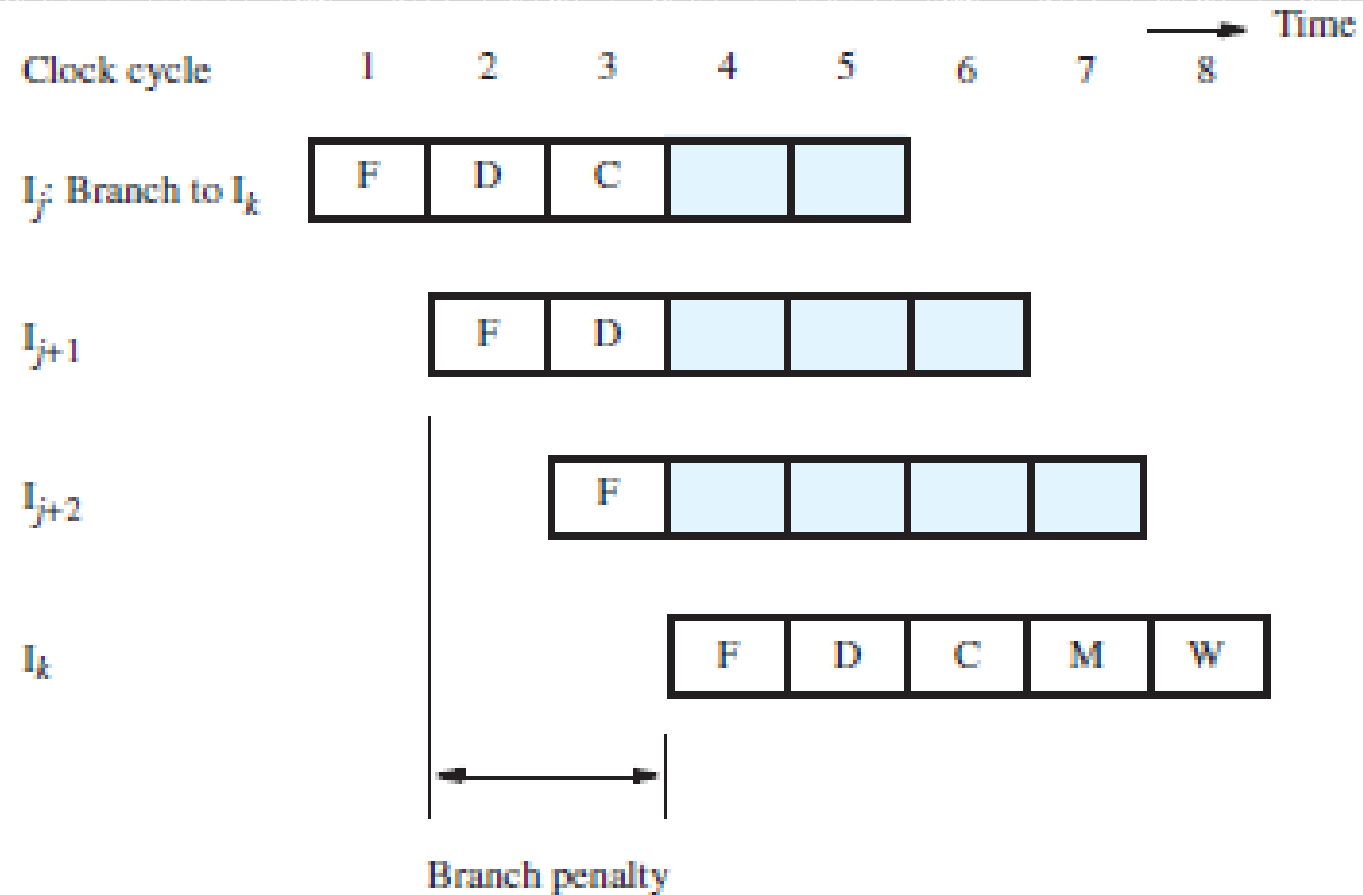


Figure 6.9 Branch penalty when the target address is determined in the Compute stage of the pipeline.

UNCONDITIONAL BRANCH

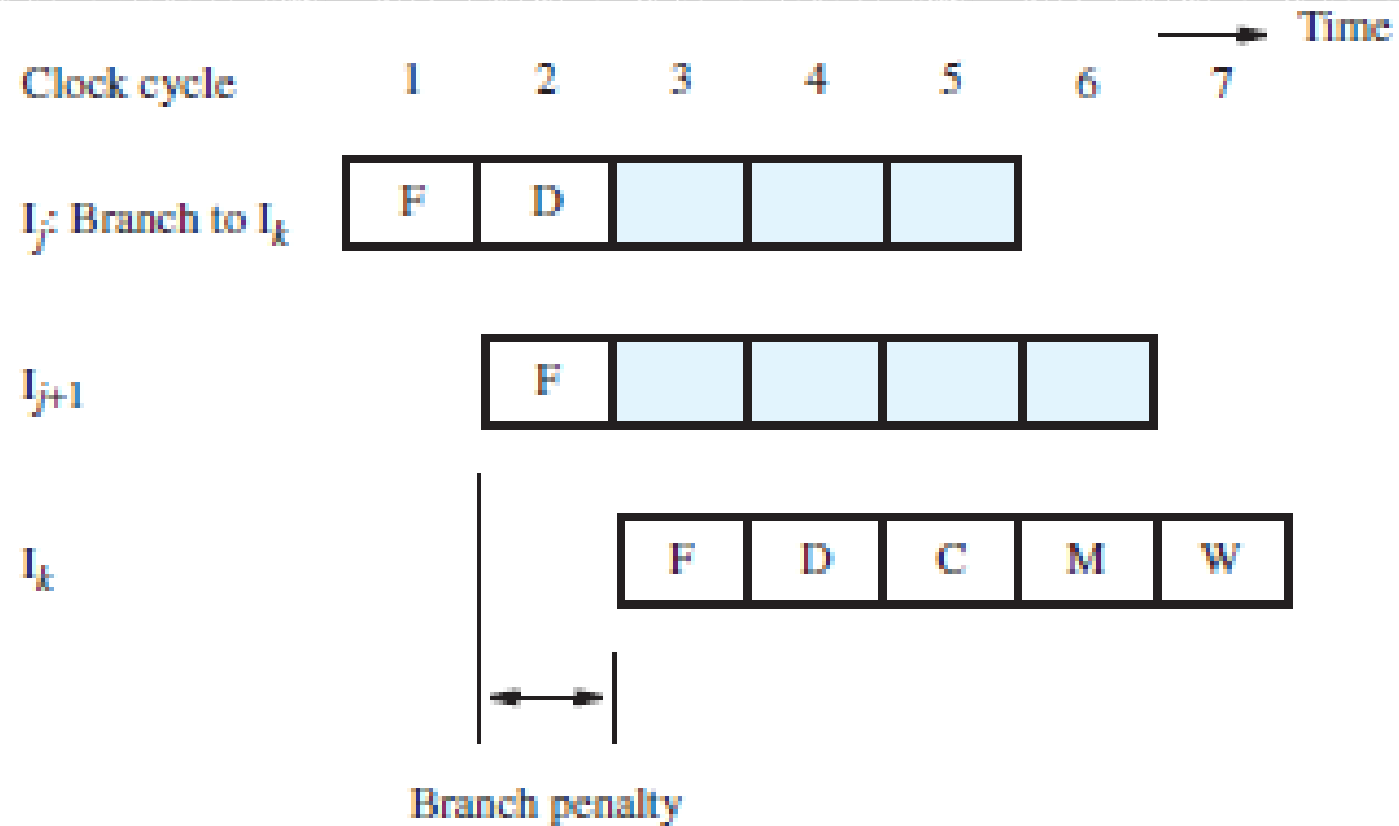


Figure 6.10 Branch penalty when the target address is determined in the Decode stage of the pipeline.

CONDITIONAL BRANCHES

Branch_if_[R5]=[R6] LOOP

- A conditional branch instruction introduces the added hazard caused by the dependency of the branch condition on the result of a preceding instruction.
- The result of the comparison determines whether the branch is taken
- The branch condition must be tested as early as possible to limit the branch penalty.
- The decision to branch cannot be made until the execution of that instruction has been completed.
- The comparator that tests the branch condition can also be moved to the Decode stage, enabling the conditional branch decision to be made at the same time that the target address is determined.
- In this case, the comparator uses the values from outputs A and B of the register file directly.

DELAYED BRANCH

- The location that follows a branch instruction is called the branch delay slot
- The instructions in the delay slots are always fetched. Therefore, arrange for them to be fully executed whether or not the branch is taken.
- Place useful instructions in these slots.
- The effectiveness depends on how often it is possible to reorder instructions.
- Branching takes place one instruction later than where the branch instruction appears in the instruction sequence. This technique is called delayed branching.

Add	R7, R8, R9
Branch_if_[R3]=0	TARGET
I_{j+1}	
\vdots	
TARGET:	I_k

(a) Original sequence of instructions containing a conditional branch instruction

Branch_if_[R3]=0	TARGET
Add	R7, R8, R9
I_{j+1}	
\vdots	
TARGET:	I_k

(b) Placing the Add instruction in the branch delay slot where it is always executed

Figure 6.11 Filling the branch delay slot with a useful instruction.

TOPICS COVERED FROM

- Textbook 1:
 - Chapter 6: 6.5- 6.6