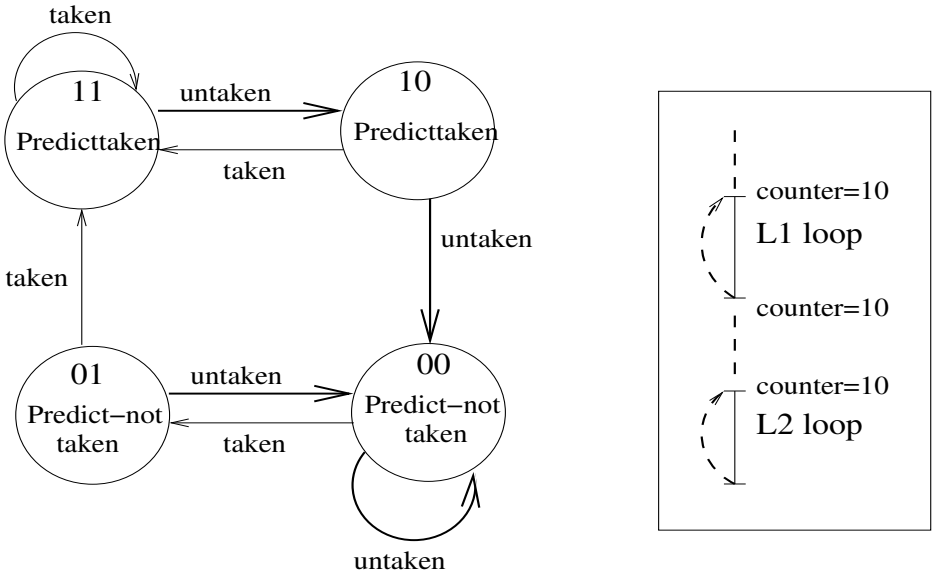
Lecture 10 August 30, 2021

Computer Architecture and Organization-II

Biplab K Sikdar

2-bit prediction Decision on prediction depends on 2-bit counter.

If count is $\geq (2^{2-1})$ -that is, 2 or 3, prediction is 'predict-taken'.



a) State transition in 2−bit prediction b) Counter initialization

Figure 14: 2-bit prediction

Counter increments in taken branch and decrements in untaken branch (Fig. 14(a)).

Before execution of a loop, 2-bit predictor is 10 (Figure 14(b)).

In 2-bit prediction, for a loop, there can be only one misprediction.

For loop with ten iterations, 9 predictions in a 2-bit prediction scheme are correct.

Its performance in this case is 90% as that of 'predict-taken' scheme.

For a loop with 100 iterations, this scheme gives 99 correct predictions (99%).

0.5.3 Branch history table

Improves prediction accuracy following recently executed branches.

Branch target buffer (BTB) also called branch history table (BHT) is used.

It is to ensure branch penalty = 0.

Branch address	Prediction bits	Target address
E_n	10	A_t
⋮	⋮	⋮

Figure 15: Branch history table

Figure 15 shows structure of a BHT. It contains

1. Address of branch instruction,
2. History bits, and
3. Probable target instruction (target address).

When a conditional branch I_b is encountered, BHT is visited.

Address of I_b is compared with branch address field of BHT.

If a match (say, *En*) is found, prediction is made based on prediction bits (10).

If prediction shows 'predict taken', target address (A_t) is taken.

If there is no match for I_b, a new record is added in BHT.

In Figure 15, a 2-bit prediction counter is considered.

We expect hit in BHT. With larger BHT, there are few misses.

0.6 Exceptions

An exception may force m/c to abort instruction in pipeline before its completion.

Exceptions can be interrupt, fault and exception.

Consider 5-stage (IF, ID,EX, MEM and WB) instruction pipeline.

Exceptions at IF can be page fault on inst fetch, memory protection violation, ...

The exception type in ID phase is: illegal opcode.

EX stage exceptions: overflow, divide by zero, floating point arithmetic anomaly, ...

The page fault on data fetch can occur in MEM phase.

However, *in WB there is no case of exception*.

Let there is a page fault for I_1 at MEM.

I_1:	IF_1	ID_1	EX_1	MEM_1	WB_1		
I_2:		IF_2	ID_2	EX_2	MEM_2	WB_2	
I_3:			IF_3	ID_3	EX_3	MEM_3	WB_3

OS will handle the page fault. Till then, if I_2, I_3, ... are in pipe.

It may result in hazards.

In a non-pipeline system, CPU cannot access I_2, I_3, ... unless page fault is resolved.

Multiple exceptions

There can multiple exceptions in the same clock cycle. Follow Figure 17.

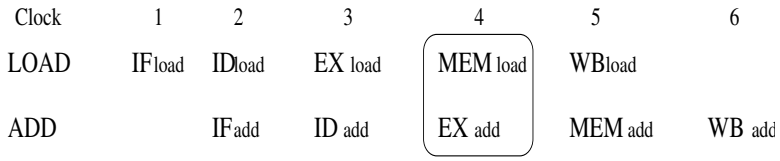


Figure 17: Multiple exceptions in same clock cycle

Out-of-order exceptions

Follow Figure 18. Exceptions are for LOAD at MEM (1) and at IF (2) for ADD.

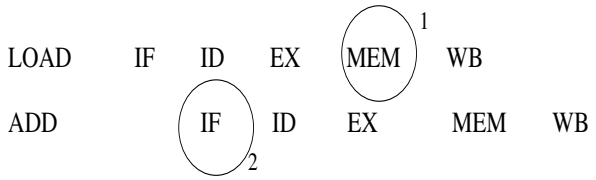


Figure 18: Out-of-order exceptions

In terms of time of event, exception (2) occurs before (1).

But (1) has to be addressed/handled prior to (2).

0.7 Dealing With Exceptions

Probable solution for handling out-of-order exceptions.

a) Each instruction I_i associates a status vector register SV_i.

   For each type of exception, there is a bit in SV_i.

   The SV_i is carried through pipeline stages along with I_i.

b) If an exception of type k is occurred, $SV_i[k]$ is set.

   Then any form of data write is turned off.

c) When I_i enters in WB or leaves MEM, SV_i is checked.

   If $SV_i[k]$ is set, exception k is handled.

0.8 Dealing With Interrupt

In a sequential processor, interrupt is acknowledge after current instruction cycle.

However, in a pipelined system, at the time of interrupt, instructions ...., $I_i - 1$, I_i, I_{i+1}, ... can be in the pipeline.

Solution 1: This scheme performs

Step 1. Stop the pipeline once an interrupt is sensed

Step 2. Complete instruction cycles of all the instructions present in the pipeline.

Step 3. Acknowledge the interrupt.

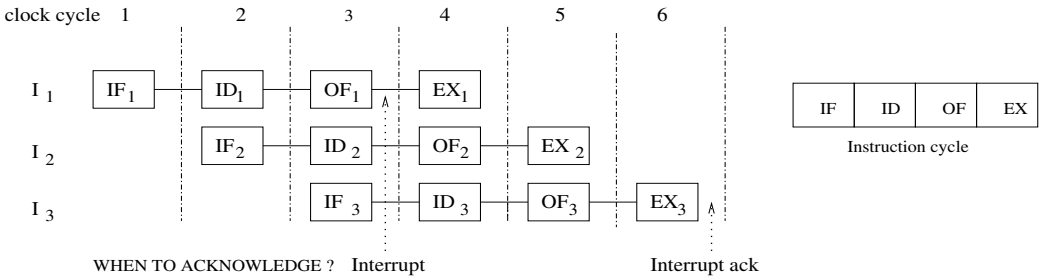


Figure 19: Interrupt handling: Solution 1

In this scheme, the interrupt service is delayed (Figure 19).

Solution 2: Flush out instructions that not reach ID phase of pipeline (Figure 20).

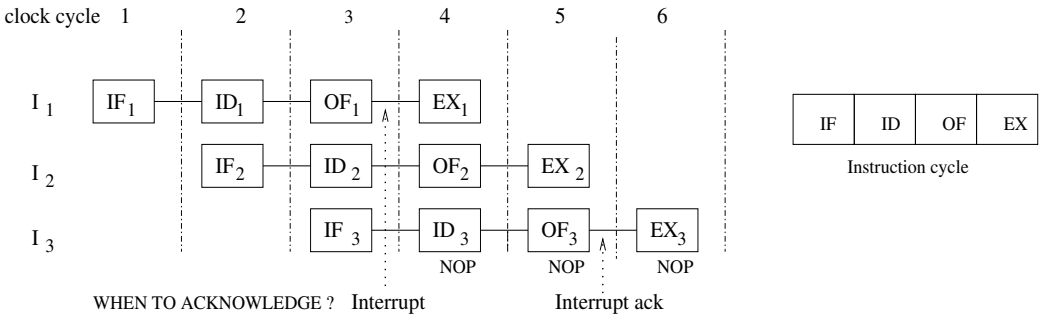


Figure 20: Interrupt handling: Solution 2

Problems

1. A program consists of two nested loops, with only single branch instruction at the end of each loop. The outer loop is executed 10 times and the inner loop 20 times. Determine the accuracy of following branch prediction strategies:

   a) always predict taken,
   b) always predict not taken,
   c) use 1-bit of history (history for each loop is initialized to ”taken”).