

Branch Dilemmas and Solutions

An important aspect of pipelining, is trying to keep a steady flow of instructions especially in the early stages of the pipeline. Conditional branches present a major hurdle for the designer.

We will discuss the following approaches in tackling the conditional branch problem:

1. Using multiple streams
2. Prefetch branch target
3. Loop buffer
4. Branch prediction
5. Delayed branch

Multiple Streams

It is possible to maintain two pipelines of initial instructions. One pipeline would contain the instructions following the branch and the other pipeline would contain instructions from the target address.

Disadvantages:

- 1) Introduces more contention delays accessing memory and registers
- 2) More branches can be introduced into the pipeline causing the need for more pipeline streams

IBM 370/168 and the IBM 3033 use two or more pipeline streams

Prefetch Branch Target

With this scheme, we automatically prefetch the target instruction along with the next instruction. The target instruction is saved until the branch is executed.

IBM 360/91 uses this method.

Loop Buffer

This is a high speed cache type memory that is used for holding the n most recently fetched instructions. It is maintained by the IF stage of the pipeline. If a branch instruction is taken, we can first check the loop buffer to see if the instruction exists. If not, we need to get it.

The loop buffer looks like the following:

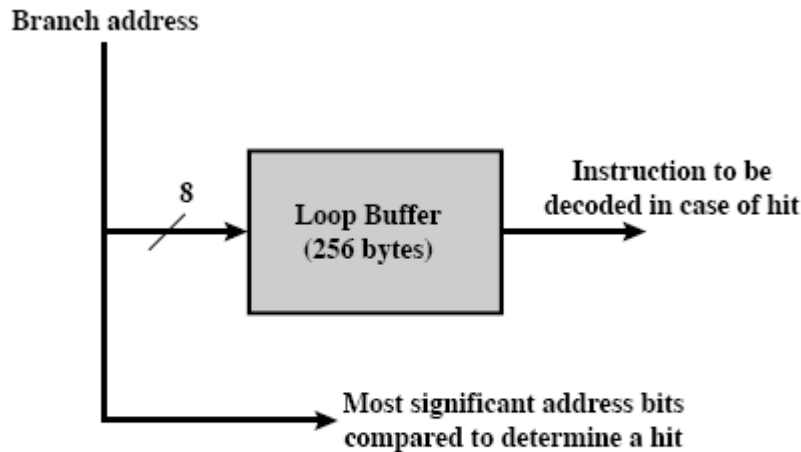


Figure 12.15 Loop Buffer

Note: The least significant bits of the branch address index the buffer while the remaining bits are used to determine if the branch target is actually in the buffer. Also, the loop buffer works just like cache only its use is dedicated to instruction fetching, it is much smaller in size, and lower in cost.

Q1: What would the branch address be if the jump below is taken?

Q2: What would the branch address be if the jump below is not taken?

13CF:0100	B80000	MOV	AX,0000
13CF:0103	BB0000	MOV	BX,0000
13CF:0106	40	INC	AX
13CF:0107	01C3	ADD	BX,AX
13CF:0109	3D0A00	CMP	AX,000A
13CF:010C	75F8	JNZ	0106
13CF:010E	90	NOP	

Benefits include:

- 1) Instructions ahead of the branch will be in the loop buffer after the first iteration of the branch.
- 2) If the branch is a few instructions ahead, which is the case with several IF-THEN type statements, the target is already in the buffer.
- 3) If the loop buffer is large enough, this strategy works well for loop type structures after the first iteration.

Branch Prediction

Several methods of branch prediction are available:

1) Predict never taken (static approach used by 68020, VAX 11/780)

Note: If the instruction following the branch would cause a page fault or protection violation, the next instruction is not fetched.

- Studies analyzing program behavior show that conditional branches are taken more than 50% of the time.
- It depends on whether we are branching forward or backward:

backward: 90% probability that it is taken

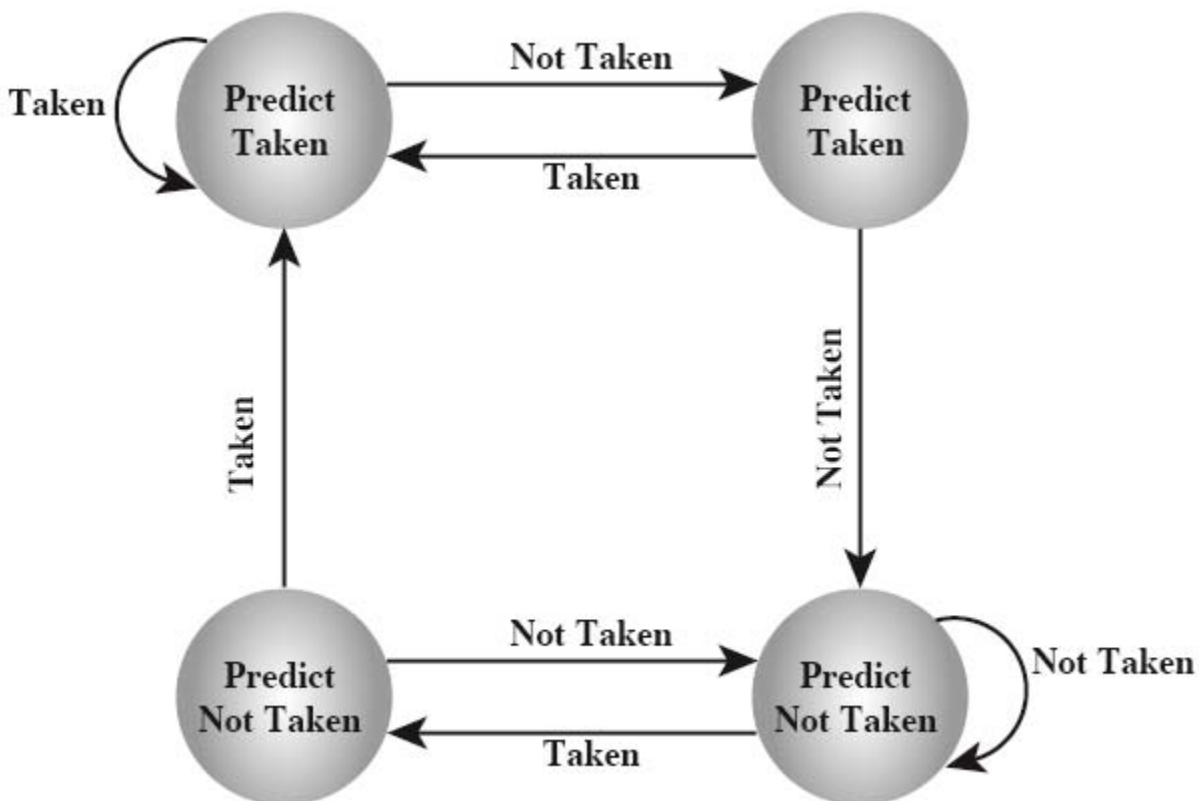
forward: IFs 50% probability taken

2) Predict always taken (static)

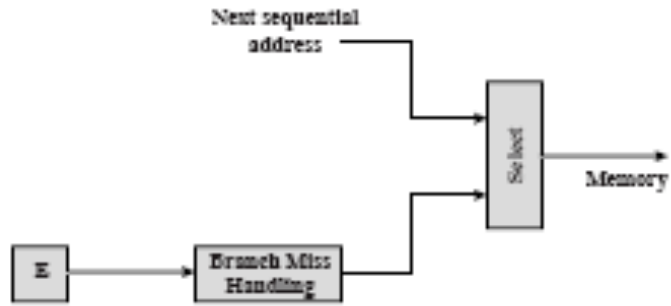
3) Predict by opcode

This approach makes a prefetch decision based on the branch's opcode. In some cases, success rates have been as high as 75%.

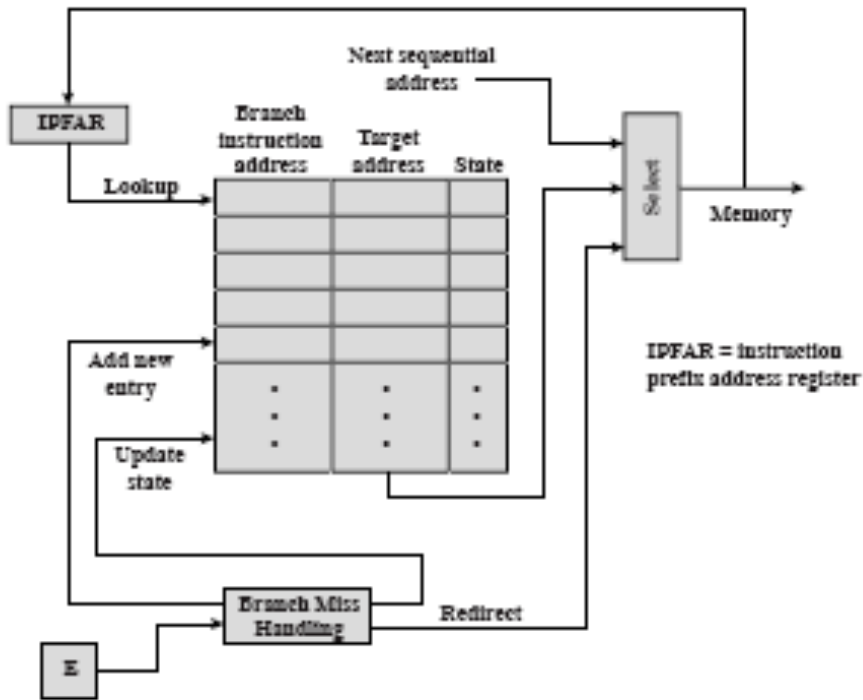
4) Switch based on taken/not taken (dynamic)



5) Branch history lookup table



(a) Predict never taken strategy



(b) Branch history table strategy

Figure 12.18 Dealing with Branches