

# Computer Architecture - tutorial 3 [TUTOR COPY]

## Context, Objectives and Organization

Supplements material from Lecture 5 (Handling hazards).

The goals of the quantitative exercises in this tutorial are: to obtain familiarity with the process of identifying hazards (E1), and to review and apply the operation of different types of branch predictors (E2).

### E1: individual – 10 mins

#### *Problem*

Consider the following MIPS code fragments, each containing two instructions. For each code fragment identify the type of hazard that exists between the two instructions and the registers involved.

a.

```
LD    R1, 0(R2)
DADD  R3, R1, R2
```

b.

```
MULT  R1, R2, R3
DADD  R1, R2, R3
```

c.

```
MULT  R1, R2, R3
MULT  R4, R5, R6
```

d.

```
DADD  R1, R2, R3
SD    2000(R0), R1
```

e.

```
DADD  R1, R2, R3
SD    2000(R1), R4
```

#### *Solution*

- a. RAW: add requires the value of R1 returned by ld
- b. WAW: add modifies the value of R1 that is also computed by mul
- c. structural hazard for multiplier
- d. RAW: sd requires (in MEM stage) the value of R1 computed by add
- e. RAW: sd requires (in ALU stage) the value of R1 computed by add

## E2: groups of 2 – 15 minutes

### Problem

- a. Explain the behaviour of a 2-bit saturating counter branch predictor. Show the state of the predictor and the transition for each outcome of the branch.
- b. Consider the following code:

```
for (i=0; i<N; i++)
  if (x[i] == 0)
    y[i] = 0.0;
  else
    y[i] = y[i]/x[i];
```

Assume that the assembly code generated is then:

```
loop: L.D    F1, 0(R2)
      L.D    F2, 0(R3)
      BNEZ   F1, else
      ADD.D  F2, F0, F0
      BEZ    R0, fall
else:  DIV.D  F2, F2, F1
fall:  DADDI  R2, R2, 8
      DADDI  R3, R3, 8
      DSUBI  R1, R1, 1
      S.D    -8(R3), F2
      BNEZ   R1, loop
```

where:

- the value of N is already stored in R1
- the base addresses for x and y are stored in R2 and R3, respectively

- register F0 contains the value 0
- register R0 (always) contains the value 0

Assuming that every other element of  $x$  has the value 0, starting with the first one, show the outcomes of predictions when a 2-bit saturating counter is used to predict the inner branch BNEZ F1, else. Assume that the initial value of the counter is 00.

*Solution*

**a. 2-bit saturating counter branch predictor**

current counter value	prediction	actual outcome	new counter value
00	NT	NT	00
00	NT	T	01
01	NT	NT	00
01	NT	T	10
10	T	NT	01
10	T	T	11
11	T	NT	10
11	T	T	11

**b. 2-bit counter prediction rate**

Iteration	current counter value	prediction	actual outcome	new counter value
1	00	NT	NT	00 (hit)
2	00	NT	T	01 (miss)
3	01	NT	NT	00 (hit)
4	00	NT	T	01 (miss)
...	...	...	...	...

2018. Vijay Nagarajan, Boris Grot, Nigel Topham and Marcelo Cintra.