# Computer Architectures
# Exam of 28.1.2020 - part I – ver. A1

*First name, Last name, ID*.....

## Question #1

Let consider the Branch Prediction Mechanism based on the Branch History Table (BHT).
You are requested to

1. Explain the reasons why a Branch Prediction Unit is important in pipelined architectures
2. Describe the general architecture and behavior of a BHT
3. Detail the architecture and behavior of a BHT using a 2-bit saturating counter
4. Detail the architecture and behavior of a BHT using a (2,2) correlating predictor.

**Hints for answers**

1. Branch prediction is important to avoid the effects of control dependencies.
   Without branch prediction, after a branch there is the risk that a wrong instruction is fetched, hence causing the need for flushing it, and thus reducing the performance.
2. A BHT is a memory composed on N words, each having m bits. Each time a branch is decoded, the log N least significant bits of its address (excluding le last ones, due to instruction alignment) are used to select a BHT entry. Based on the result of the branch instructions referring to the entry, its content is used for the prediction. When the branch result is known, the BHT entry is possibly updated.
3. In the case of a 2-bit saturating counter BHT, each entry of the BHT corresponds to a 2-bit saturating counter. This counter is incremented each time the branch is taken, decremented elsewhere. The counter is not updated in case its value is 0 and the branch is not taken, or if it is 3 and the branch is taken. When its value is 01 or 1, the branch is predict not taken, if it is 2 or 3, it is predicted taken.
4. In this case each BHT entry corresponds to 4 2-bit counters. During the program execution, a 2-bit shift register is continuously updated by shifting in a 1 each time a taken branch is executed, a 0 each time a not taken branch is executed. When a branch is decoded, an entry in the BHT is selected, and the field (out of the four) is considered, given by the shift register value. Then, this field is used as in the previous point.

# Computer Architectures
# Exam of 28.1.2020 - part I – ver. A1

*First name, Last name, ID*.....

## 5. Question #2

Let consider a MIPS64 architecture including the following functional units (for each unit the number of clock periods to complete one instruction is reported):

- Integer ALU: 1 clock period
- Data memory: 1 clock period
- FP arithmetic unit: 2 clock periods (pipelined)
- FP multiplier unit: 4 clock periods (pipelined)
- FP divider unit: 8 clock periods (unpipelined)

You should also assume that

- The branch delay slot corresponds to 1 clock cycle, and the branch delay slot is not enabled
- Data forwarding is enabled
- The EXE phase can be completed out-of-order.

You should consider the following code fragment and, using the table in the following page (where each column corresponds to a clock period), and determine the pipeline behavior in each clock period, as well as the total number of clock periods required to execute the fragment, reporting the result in the right column in the table below. The value of the constant k is written in f5 before the beginning of the code fragment.

```
; ********************* MIPS64 ***********************
;  for (i = 0; i < 10; i++) {
;       v4[i] = v1[i]/v2[i] + v3[i]*k;
;  }
```

```
                .data
V1:     .double "10 values"
V2:     .double "10 values"
V3:     .double "10 values"
V4:     .double "10 values"



                .text
main:  daddui r1,r0,0
       daddui r2,r0,10
loop:  l.d  f1,v1(r1)
       l.d  f2,v2(r1)
       l.d  f3,v3(r1)
       mul.d  f5,f3,f4
       div.d  f6, f1, f2
       add.d f7, f6, f5
       s.d  f7,v4(r1)
       daddui  r1,r1,8
       daddi  r2,r2,-1
       bnez  r2,loop
       halt
```

| | comments | Clock cycles |
|---|---|---|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | r1← pointer | 5 |
| | r2 <= 20 | 1 |
| | f1 <= v1[i] | 1 |
| | f2 <= v2[i] | 1 |
| | f3 <= v3[i] | 1 |
| | f5 <= v3[i]*k | 5 |
| | f6 <= v1[i] / v2[i] | 5 |
| | f7 <= v3[i]*k + v1[i] / v2[i] | 2 |
| | v4[i] <= f7 | 1 |
| | r1 <= r1 + 8 | 1 |
| | r2 <= r2 - 1 | 1 |
| | | 2 |
| | | 1 |
| total | | 216 |

# Computer Architectures
# Exam of 28.1.2020 - part I – ver. A1

*First name, Last name, ID*.....

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| daddui r1,r0,0 | F | D | E | M | W | | | | | | | | | | | | | | | | | | | | | | | | | | |
| daddui r2,r0,10 | | F | D | E | M | W | | | | | | | | | | | | | | | | | | | | | | | | | |
| l.d f1,v1(r1) | | | F | D | E | M | W | | | | | | | | | | | | | | | | | | | | | | | | |
| l.d f2,v2(r1) | | | | F | D | E | M | W | | | | | | | | | | | | | | | | | | | | | | | |
| l.d f3,v3(r1) | | | | | F | D | E | M | W | | | | | | | | | | | | | | | | | | | | | | |
| mul.d f5,f3,f4 | | | | | | F | D | | E | E | E | E | M | W | | | | | | | | | | | | | | | | | |
| div.d f6, f1, f2 | | | | | | | F | | D | E | E | E | E | E | E | E | E | M | W | | | | | | | | | | | | |
| add.d f7, f6, f5 | | | | | | | | | F | D | | | | | | | E | E | M | W | | | | | | | | | | | |
| s.d f7,v4(r1) | | | | | | | | | | F | | | | | | | D | | E | M | W | | | | | | | | | | |
| daddui r1,r1,8 | | | | | | | | | | | | | | | | | F | | D | E | M | W | | | | | | | | | |
| daddi r2,r2,-1 | | | | | | | | | | | | | | | | | | | F | D | E | M | W | | | | | | | | |
| bnez r2,loop | | | | | | | | | | | | | | | | | | | | F | | D | E | M | W | | | | | | |
| halt | | | | | | | | | | | | | | | | | | | | | | F | D | E | M | W | | | | | |