**Question #1**

The Tomasulo architecture for superscalar processors with dynamic scheduling and speculation uses reservation stations.

You are requested to

* Explain what reservation stations are and where they are placed in the Tomasulo architecture, listing the modules they are connected to
* Describe the hardware structure of a reservation station
* Summarize when data/information are written/updated in a reservation station.

Reservation stations are buffers that buffer the operands of instruction waiting to issue, operands are stored in reservation station as soon as they are available. They are placed before the functional units. Reservation stations relate to functional unit, CDB, instruction queue and file register.

Reservation station is composed of: Op operation to be performed, Vjk operands that are already available, Qjk operands that aren’t available, A address used only in store/load buffer, Busy shows if the reservation is free or not.

The reservation stations are written when an instruction is issued, and when in the CDB there is a result operator that is needed by Reservation station.

**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: 6 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, filling the following tables, determine the pipeline behavior in each clock period, as well as the total number of clock periods required to execute the fragment. The value of the constant k is written in f10 before the beginning of the code fragment.

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* MIPS64 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; for (i = 0; i < 10; i++) {

; v5[i] = (v1[i]\*v2[i]) + (v3[i]\*v4[i])/k;

; }

|  |  |  |
| --- | --- | --- |
| .data | Comments | Clock cycles |
| v1: .double “10 values” |  |  |
| v2: .double “10 values” |  |  |
| v3: .double “10 values”  v4: .double “10 values”  v5: .double “10 values” |  |  |
|  |  |
|  |  |
|  |  |
| .text |  |  |
| main: daddui r1,r0,0 | r1← pointer | 5 |
| daddui r2,r0,10 | r2 <= 20 | 1 |
| loop: l.d f1,v1(r1) | f1 <= v1[i] | 1 |
| l.d f2,v2(r1) | f2 <= v2[i] | 1 |
| l.d f3,v3(r1) | f3 <= v3[i] | 1 |
| l.d f4,v4(r1) | f4 <= v4[i] | 1 |
| mul.d f6,f1,f2 | f6 <= v1[i]\*v2[i] | 4 |
| mul.d f7,f3,f4 | f7 <= v3[i]\*v4[i] | 1 |
| div.d f8, f7, f10 | f8 <= v3[i]\*v4[i]/k | 6 |
| add.d f9, f6, f8 | f9 <= v1[i]\*v2[i] + v3[i]\*v4[i]/k | 2 |
| s.d f9,v5(r1) | v5[i] <= f9 | 1 |
| daddui r1,r1,8 | r1 <= r1 + 8 | 1 |
| daddi r2,r2,-1 | r2 <= r2 - 1 | 1 |
| bnez r2,loop |  | 2 |
| halt |  | 1 |
| total |  | 236 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| main: daddui r1,r0,0 | F | D | E | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| daddui r2,r0,10 |  | F | D | E | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| loop: 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 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| l.d f4,v4(r1) |  |  |  |  |  | F | D | E | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| mul.d f6,f1,f2 |  |  |  |  |  |  | F | D | E | E | E | E | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| mul.d f7,f3,f4 |  |  |  |  |  |  |  | F | D | E | E | E | E | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| div.d f8, f7, f10 |  |  |  |  |  |  |  |  | F | D |  |  |  | E | E | E | E | E | E | M | W |  |  |  |  |  |  |  |  |  |  |  |  |  |
| add.d f9, f6, f8 |  |  |  |  |  |  |  |  |  | F | D |  |  |  |  |  |  |  |  | E | E | M | W |  |  |  |  |  |  |  |  |  |  |  |
| s.d f9,v5(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 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |