Rs\_memory.v

Inputs:

Added the rob\_slot which is an input to maintain what ROB slot the data from the reservation station corresponds to.

Outputs:

Added the output register rob\_dest which is a value that specifies which rob slot the output data corresponds to.

Implementation:

Adding register files to hold information corresponding to the rob slot for each line in the reservation station. Rob\_dest getting updated when a slot in the reservation station is sent to the memory unit.

Rs\_adders.v

Inputs:

Added the rob\_slot which is an input to maintain what ROB slot the data from the reservation station corresponds to.

Outputs:

Added the output register rob\_dest which is a value that specifies which rob slot the output data corresponds to.

Implementation:

We also added a register array “reg [1:0] rob [0:2]:” that holds data to store what each line in the particular reservation corresponds to in the reorder buffer. We have the output rob\_dest being set in our “output data” section of the module. It gets updated according to which spots in the reorder buffer are currently free and able to be loaded with more data.

Rs\_load.v

Inputs:

Added the rob\_slot which is an input to maintain what ROB slot the data from the reservation station corresponds to.

Outputs:

Added the output register rob\_dest which is a value that specifies which rob slot the output data corresponds to.

Implementation:

We also added a register array “reg [1:0] rob [0:2]:” that holds data to store what each line in the particular reservation corresponds to in the reorder buffer. Add more here

Rs\_store.v

Inputs:

Added the rob\_slot which is an input to maintain what ROB slot the data from the reservation station corresponds to.

Outputs:

Added the output register rob\_dest which is a value that specifies which rob slot the output data corresponds to.

Implementation:

We also added a register array “reg [1:0] rob [0:2]:” that holds data to store what each line in the particular reservation corresponds to in the reorder buffer. Add more here

Rs\_multipliers.v

Inputs:

Added the rob\_slot which is an input to maintain what ROB slot the data from the reservation station corresponds to.

Outputs:

Added the output register rob\_dest which is a value that specifies which rob slot the output data corresponds to.

Implementation:

We also added a register array “reg [1:0] rob [0:2]:” that holds data to store what each line in the particular reservation corresponds to in the reorder buffer. We have the output rob\_dest being set in our “output data” section of the module. It gets updated according to which spots in the reorder buffer are currently free and able to be loaded with more data.

Adder (within tomasulo.v)

Inputs:

Added the rob\_slot input to the adder within the tomasulo module. This input contains which ROB slot corresponds to which line of data in the adder reservation station.

Outputs:

Added the rob\_dest output to the adder within the tomasulo module. This output is a value that specifies which rob slot the output data of the given reservation unit corresponds to, it is a pair with rob\_slot.

Implementation:

This can be implemented by just adding the given input and output to the adder reservation station unit within the tomasulo.v module.