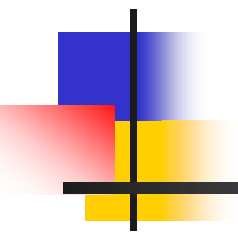
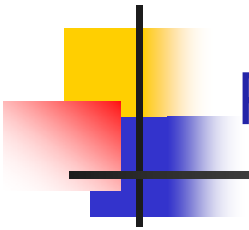
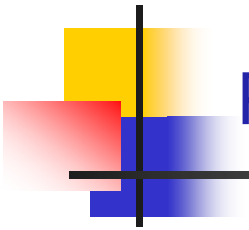


ISSUE QUEUES
(Integer, Multiplier and Divider)



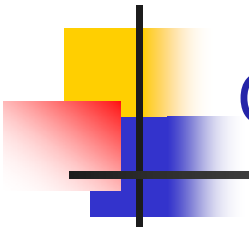
FEATURES OF ISSUE QUEUE

- Depth of Issue queue : 8
- Unlike the Summer 2009 Tomasulo, the new issue queues carry just the tag and not the data of Rs, Rt, Rd registers.
- Here, Physical register addresses of Rs, Rt, Rd are the tags.
- Integer Issue queue supports jal/jr, Addi instructions.



FEATURES OF ISSUE QUEUE

- All Issue queues looks at the output of CDB, ALU unit, 3rd stage of Mult execution unit and output of Div execution unit for ready signals of Rs, Rt registers.
- Also, in Int Issue queue, JR rs instruction is given higher priority.
- Issue queue is flushed when CDB_Flush signal is generated and instruction is younger to the instruction on CDB – *selective flushing*.



Operations in Issue Queue

- Every new instruction dispatched is given a slot in the queue.
- Every issued instruction shifts other instructions in queue by one location – *shift update*.
- If 2 instructions are ready at same time, priority is given to older instruction only except JR $rs instruction (*which has the highest priority*).
- Each instruction is ready to be issued only when the required registers(Rs, Rt) are ready and instruction is valid.

Fields in each Issue Queue Registers

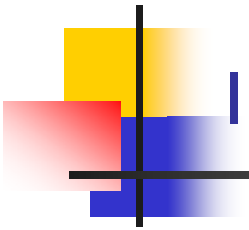
ROBTag (5 bits)	Rs(6 bits)	RsRdy Bit	Rt(6bits)	RtRdy Bit	Opcode	Rd(6 bits)	Instr. Valid bit	Reg. Write

- The Opcode field is 3 bits and required only by Int Issue Queue as ALU needs it to identify the operation to be performed.
- Multiply and Divide execution look at the Valid bit of Instruction to perform any operation. Hence, Opcode bits are of no significance in these queues.

Additional Fields in Int. Issue Queue Registers

Immediate (16 bits)	Branch bit	Branch Predict bit	Branch Other Addr (32 bits)	Branch PCBits (3 bits)	JR Valid bit	JR $31 Valid bit	JAL Valid Bit

- Immediate field carries the value for Addi instruction.
- Branch PC Bits are used for updating the Branch Prediction Buffer.
- Branch Other Addr is the Addr to which it needs to jump if mispredicted.
- Branch Predict Bit is to identify whether branch was taken or not taken.



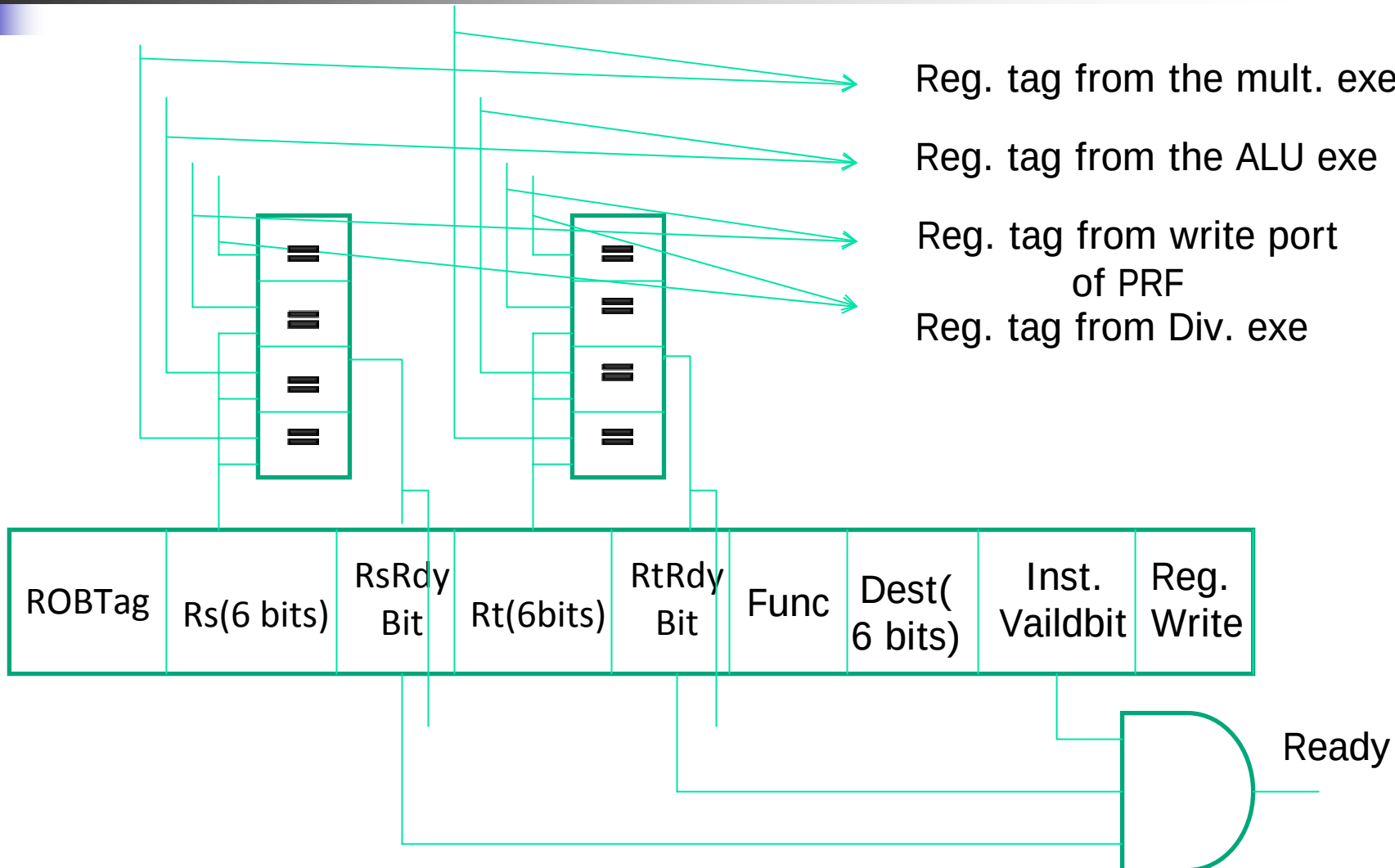
Issue Queue: Concept of Rs and Rt Ready Signal

The Rs and Rt Ready Signals are dependent on the following:

1. Instruction Valid Signal.
2. Rt or Rs Ready signal.
3. Rd tag and Reg Write signal of CDB.
4. Rd tag and Ready signal of 3rd stage of Mult Unit.
5. Rd tag and Ready signal of Div Unit.
6. Rd tag and RegWrite Signal of ALU Unit*.

Note: *The Integer Issue queue differs from other queues because it has to look at it own output for the 6th condition given above.

Analysis of each instruction READY Bit



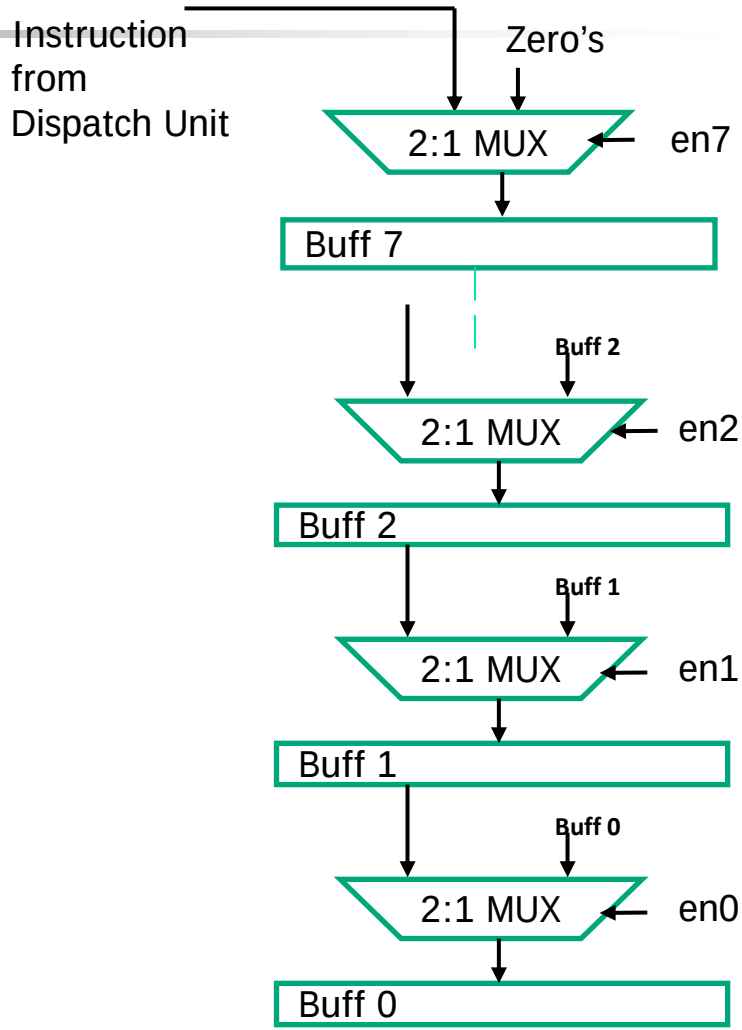
Issue Queue Controller: Concept of Shift update

- The En signal is generated based on following:

1. Instruction is Valid
2. Instruction is JAL or

   Rs is Ready and,

   1. Rt is Ready or
   2. Instruction is JR or
   3. JR rs



Interface with the Dispatch Unit.

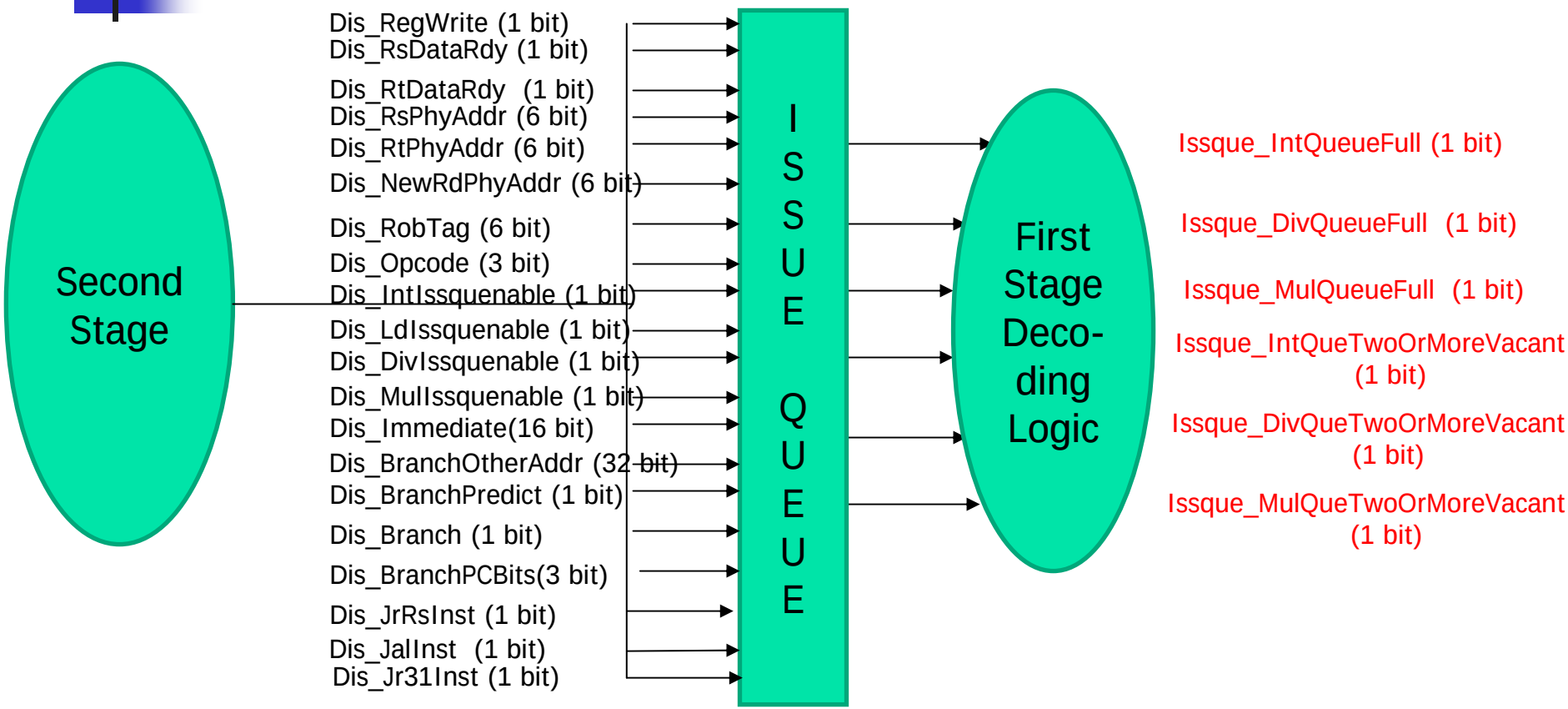
- Since, the dispatch is 2-stage when need following signals:
  - **Full Signal**: means all slots occupied.

    Reason: To stall dispatching on instructions to issue queue.
  - **Two or More Slots Vacant Signal**: means 2 or more slots are available for dispatch to fill.

    Reasons:

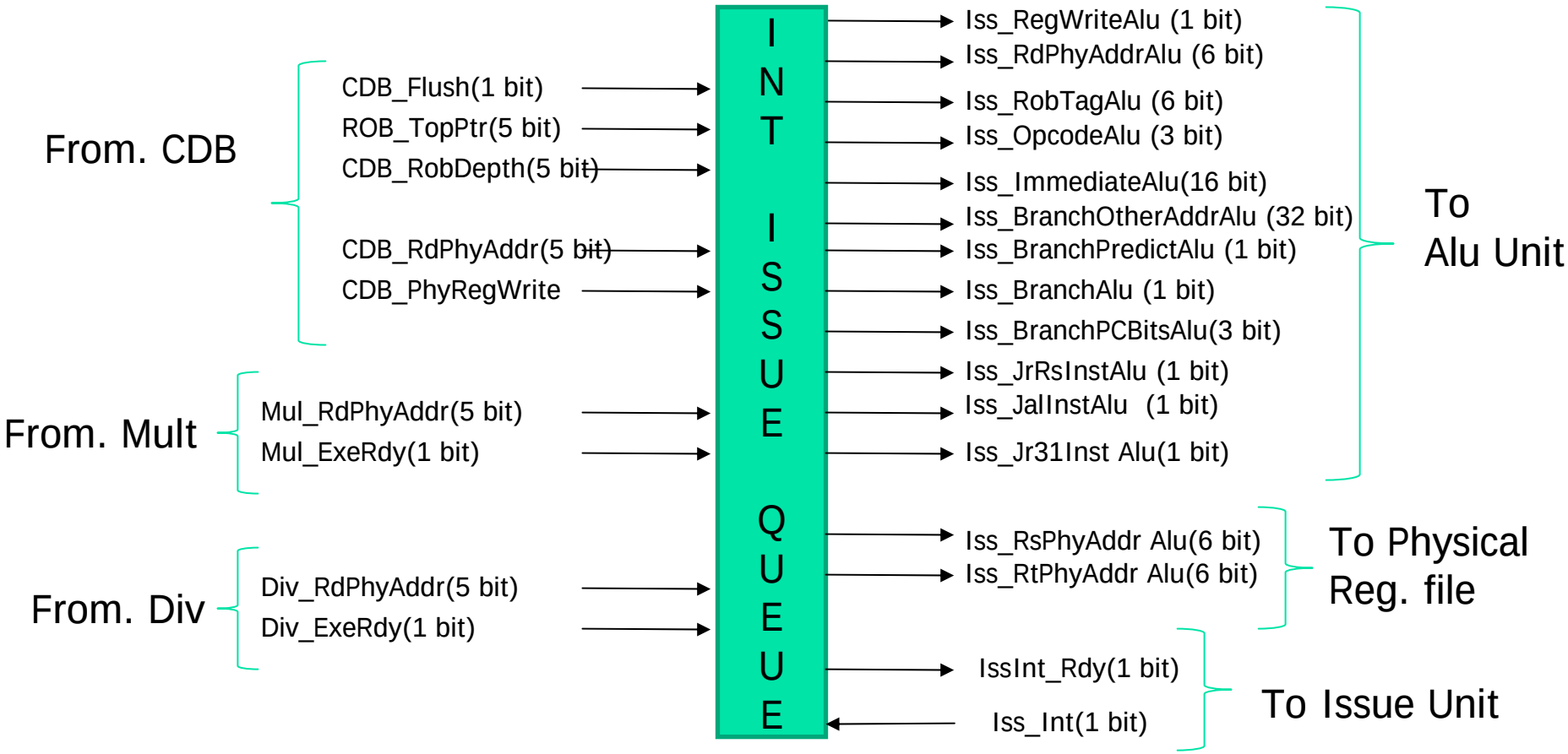
1. Only one slot vacant in queue: The instruction in first stage cannot be issued by dispatch.
2. Two or more slots vacant in queue: The instruction in first stage of dispatch finds a slot in queue.

Interface with the Dispatch Unit

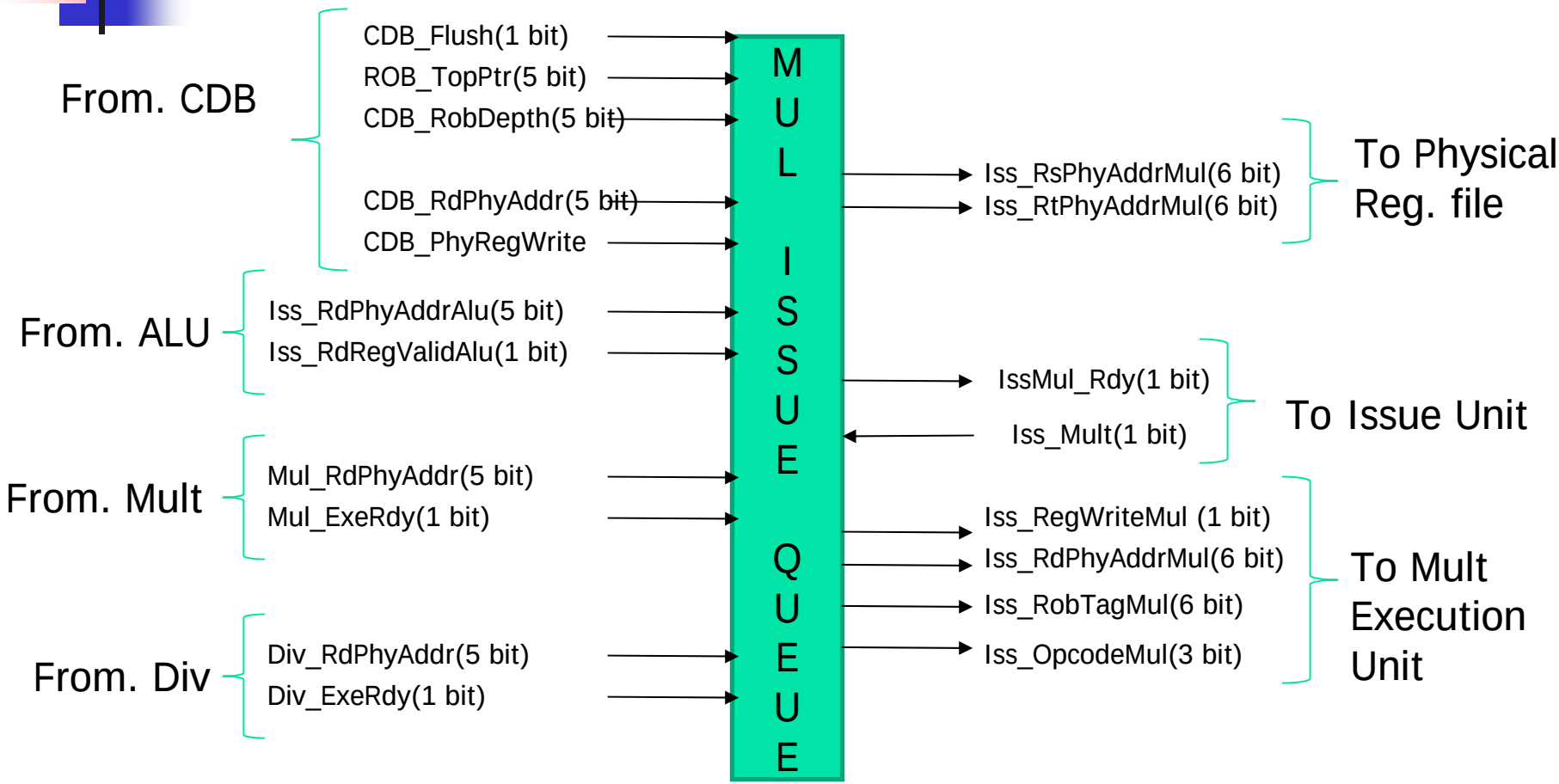


NOTE: Integer issue queue needs to distinguish between jr , jal and rest of the instructions. Thus **Dis_JrRsInst** (for jr $rs) **, Dis_JalInst** (for jal) and **Dis_JrInst** (for jr $31) are provided.

Interface with Execution units, Physical Register File and CDB.



Interface with Execution units, Physical Register File, Issue Unit and CDB.



Interface with Execution units, Physical Register File, Issue Unit and CDB.