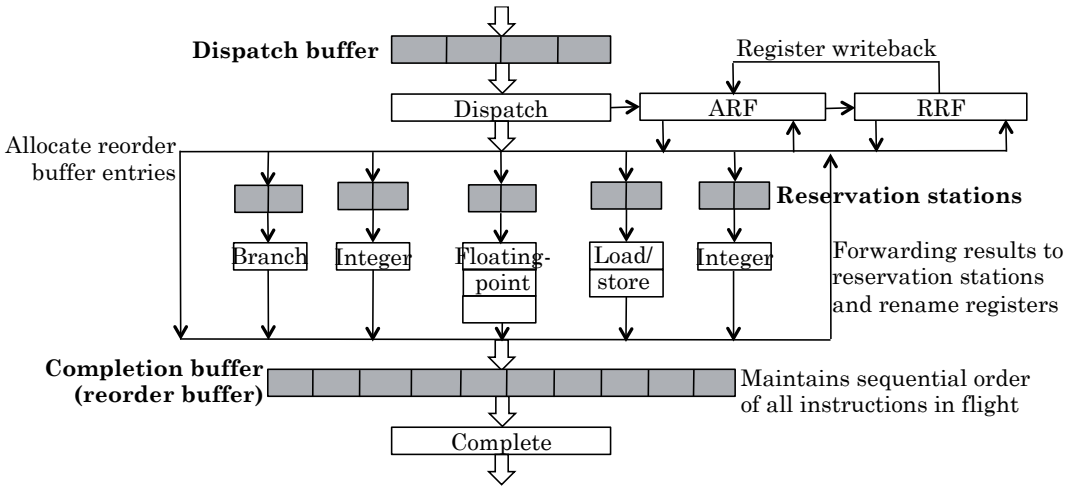
# CS6600: Computer Architecture
## Assignment #5

Submission deadline: 14th Oct, 2021

Implement the Dynamic Execution Core as shown below:




The dispatch buffer consists of decoded instructions. In every cycle, at most *n* (i.e., decode width) decoded instructions are added to the dispatch buffer, based on the availability of space in the dispatch buffer. Before dispatching the decoded instructions into respective reservation stations, register renaming must be done. We consider distributed reservation station concept here. The simulator must implement out-of-order execution of ALU instructions, in-order execution of memory instructions, and operand forwarding. To simplify the things, let us assume that there is no branch prediction unit implemented in the processor. Assume that both instruction cache and data cache are perfect, meaning, all the requests to them will be hit. To support in-order commit, implement reorder buffer. Include the values for any of the parameters (such as decode width, issue width, commit width, functional unit latencies, cache access latencies, buffer entries, etc.) required by the simulator in a configuration file. The simulator has to report the total number of cycles required for executing a given set of instructions.

Instruction Format: 16 bit – instructions.

| OPCODE (4-bits) | R1 (4-bits) | R2 (4-bits) | R3 (4-bits) |
|---|---|---|---|

R1 – Destination register
R2, R3 - Source registers
16 physical registers and 16 rename registers are available.

Opcodes:

| | |
|---|---|
| Add | 0001 |
| Sub | 0010 |
| Mul | 0011 |
| Div | 0100 |
| Load | 0101 |
| Store | 0110 |

For load/store, the address will be given in two parts a base register and a 4-bit offset.

LD R1, R2, Offset – {R1<- R2+Offset}
ST R1, R2, Offset – {R1 -> R2+Offset}

There are 3 functional units, adder/subtractor, multiplier and division FU along with a load/store unit.

Inputs:

1. Text file that contains a list of instructions (File name will be supplied, ex: ins.txt)
2. Issue width (n) – Instructions moved from decode queue to reservation stations in a cycle.
3. Latencies for the 3 FUs.
4. No. of entries in the reservation station of all the 4 units.
5. No. of entries in the ROB.
6. Cache access latency.

Output:

1. No. of cycles required to execute the given list of instructions.