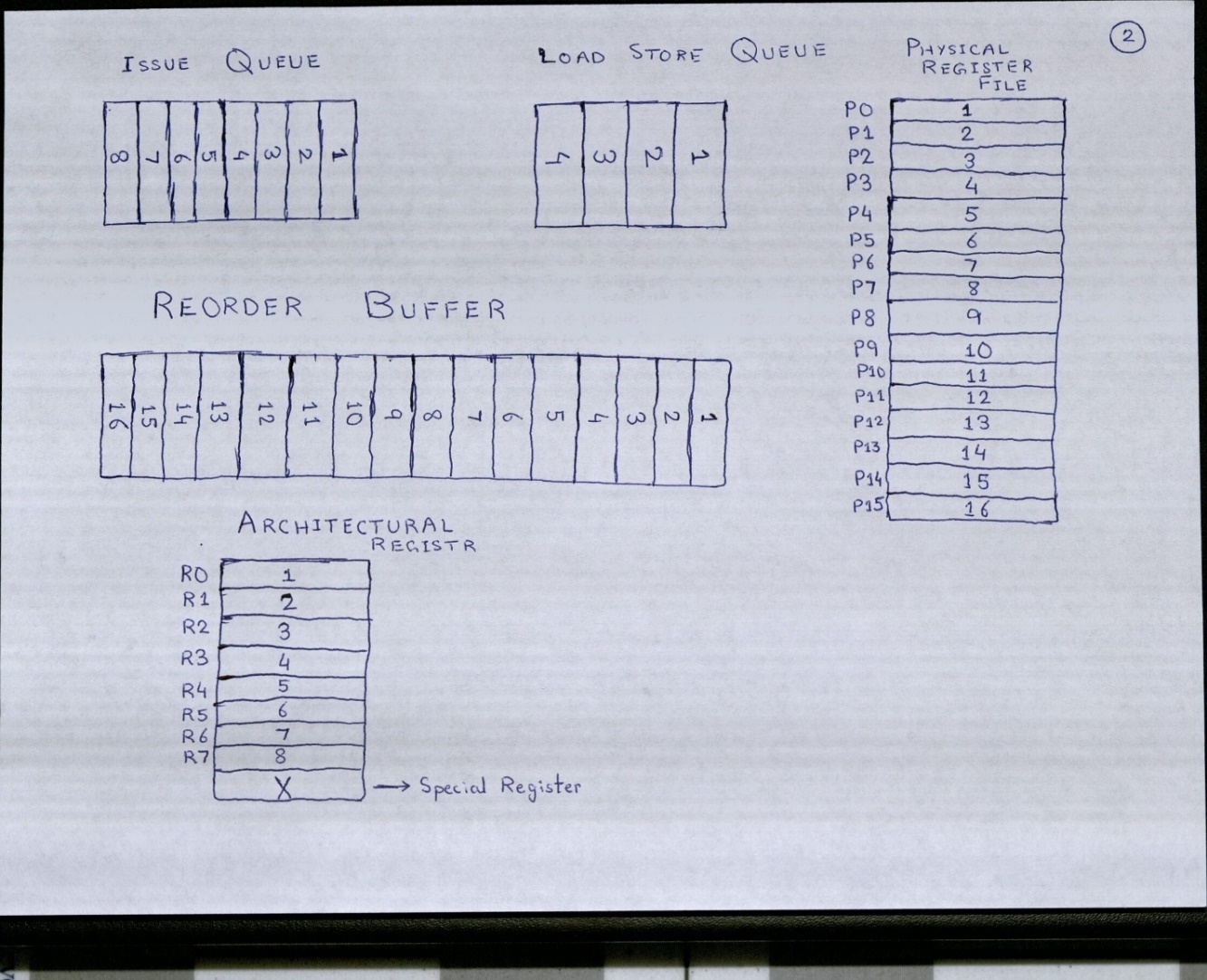
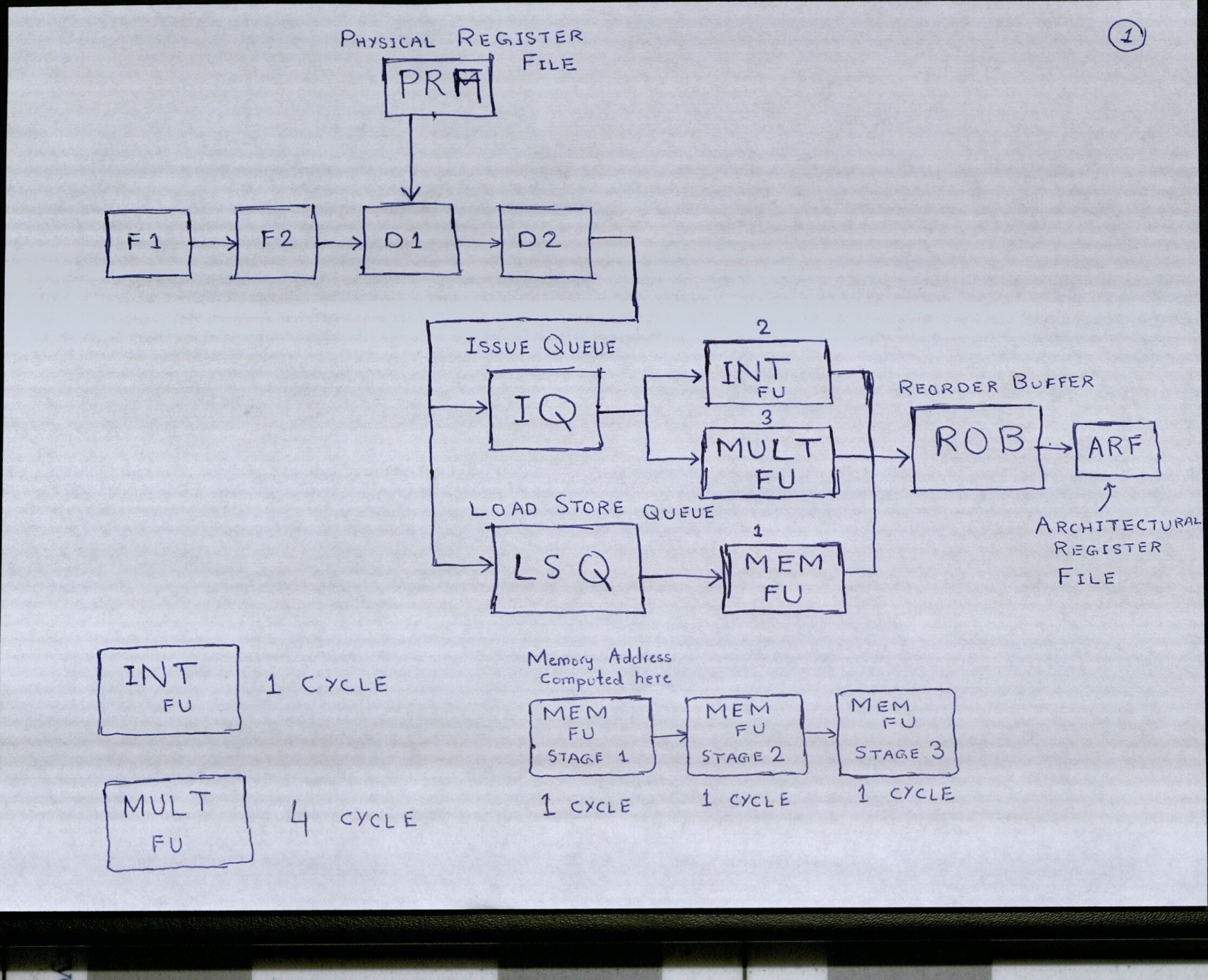
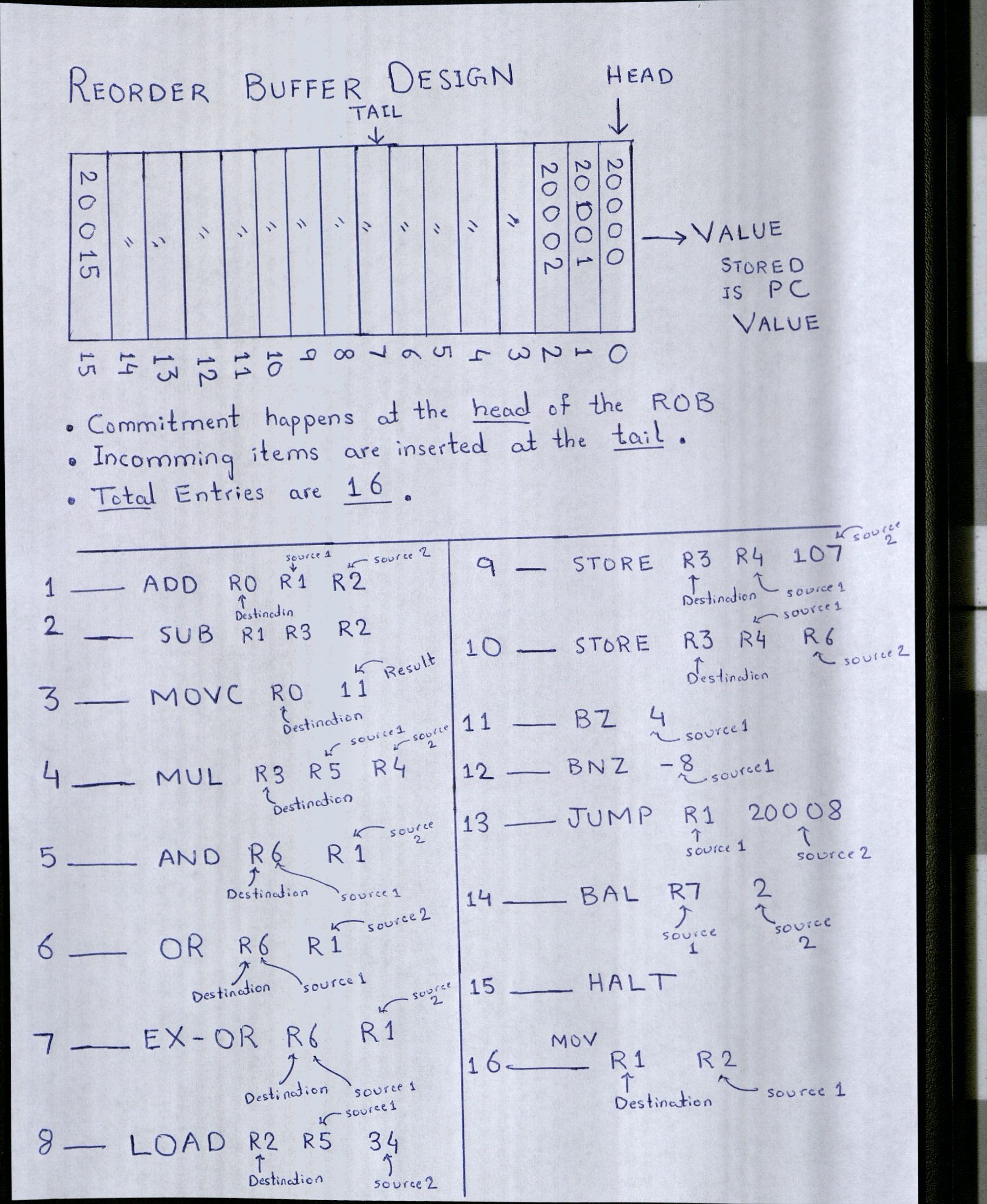
|  |  |
| --- | --- |
| **Fixed Size Queue** | **ROB (Circular Queue)** |
| **FixedQueue(int x):** *Constructor*  **void add(int x):** *Add an element to the queue*  **int remove():** *Remove an element from the queue*  **int see():** *Peak at the element at the head of the queue*  **int[] getElements():** *get all the elements of in the queue in an array*  **boolean isFull():**  **boolean isEmpty():** | **ReorderBuffer(int x):** *Constructor*  **int getHead():** *returns the element at the head of the Queue*  **int getTail():** *get the item at the tail*  **void insert(int x):** *insert element at the tail of the ROB*  **void commit():** *commit the instruction at the head of the tail*  **boolean isFull():**  **boolean isEmpty():** |

**INSTRUCTION FORMAT**

|  |  |  |
| --- | --- | --- |
| 0 | **PC** Value | Starts at 20000 |
| 1 | **STAGE** of Execution | 0 – INITIAL STAGE  1 – FETCH STAGE 1  2 – FETCH STAGE 2  3 – DECODE STAGE 1  4 – DECODE STAGE 2  5 – QUEUE (IQ or LSQ)  6 – FU STAGE  7 – ROB WAITING FOR COMMITMENT  0 – FINISHED EXECUTION |
| 2 | **INSTRUCTION TYPE** | ADD, SUB, MUL, or BAL |
| 3 | **FU TYPE** | Int FU, Mult FU, or Mem FU |
| 4 | **CYCLES** | The number of cycles the instruction will spend in the FU |
| 5 | **PHY REGISTER 1 (ARC REGISTER 1)** | The Physical register for Architectural Register 1 to read the source value. -1 means, get the value directly from the Architectural Register |
| 6 | **PHY REGISTER 2 (ARC REGISTER 2)** | The Physical register for Architectural Register 2. |
| 7 | **DATA (SOURCE 1)** |  |
| 8 | **DATA (SOURCE 2)** |  |
| 9 | **RESULT VALUE** | -1 means result invalid |
| 10 | **SOURCE REGISTER 1** |  |
| 11 | **SOURCE REGISTER 2** |  |
| 12 | **DESTINATION REGISTER** |  |





**RUNNING THE SIMULATOR**

* **initialize** – this resets the register values to zero, resets values stored at memory locations to 0, sets PC (Program Counter) to point to the first instruction, and sets the cycles to zero.
* **display** – this will display the contents of the pipeline, the contents of the Register, the contents of the memory, current value of the PC (Program Counter) and the number of cycles processed.
* **simulate <n>** - this will simulate n cycles and pause the simulator.