# Data Path Components:

## Instruction Fetch

1. PC : Stores the PC value of the current instruction.
2. IM : Instruction Memory from which the instruction is obtained.
3. PC++ : It increments the PC value by 1 for the next possible instruction.
4. Brach Prediction Table Block : Store the already encountered BEQ instruction PCs and the branch address along with the only more recent history of the instruction (whether it was taken or not)
5. MUX : Chooses between the taken address given by the branch prediction table and PC++. The control signal is also generated by the table block itself

## Instruction Decoder

1. SE : It sign extends the 6-bit or 9-bit immediate data to 16 bit. It also observes the extra LLI bit in which case it only packs the immediate data by zeros.
2. SE+PC : It adds the SE value and the PC value of the current instruction.
3. Decoder: It gives the control signals according to the current instruction.
4. MUX : Steers input between PC+1 and (SE+PC) value. Chooses (SE+PC) when JAL instruction.

## Register Read

1. LS : Left shifts the sign extended value.
2. RF : Register file which contains the register (R0-R7).
3. MUX1 : It steers inputs between (SE+PC) and LS. SE+PC is chosen during JAL instruction and LS is chosen during LHI instruction.
4. MUX2 : The mux before AR2. It steers inputs between AR2 from IDRR register and the output from LM\_SM block.
5. MUX3 : The mux after DO1. It steers inputs between DO1 and the ALU output which gives the DO+1 value during LM\_SM instruction.
6. Hazard Mux : Controlled by Hazard\_RR.
7. Hazard\_RR : The Hazard Block present in RR stage. Sends Input to the PC register according to the instruction.
   1. SE during LLI and R7 is the destination.
   2. SE\_LS during LHI and R7 is the destination.
   3. DO1 during the JLR instruction.
8. LM\_SM mux : Choses the input for the LM\_SM block between the data after decoder or from the ID\_RR pipeline.
9. LM\_SM block : It operates the LM and SM operation.

## Execution

1. ALU : Perform ADD, NAND and Comparator operation.
2. Flags : Contains the carry, zeros and overflow flags.
3. Forwarding Blocks : Checks if dependency present between RR\_EX pipeline and the pipeline registers in the further stages. It accordingly controls the forward logic muxes.
4. MUX1 : The second mux before ALU second input. It steers inputs according to the arithmetic operation or address calculation.
5. MUX2 : The mux just before the second input of ALU. Controlled by the LM\_SM block. Sends 1 during the LM or SM operation to increment the address.
6. Hazard Mux : Mux controlled by the hazard logic block in execution stage.
7. Staller : It stalls the pipeline registers when there is an immediate dependency after the load instruction.
8. Hazard\_EX : Controls the hazard when during arithmetic instruction the destination is R7. It loads the PC with the required value controlling the hazard mux. Flushes the pipeline.

## Memory Write/Read

1. MEM : Data memory from which the instruction read data or writes data.
2. MUX1 : Mux before address of data memory. Steers inputs between ALU output (address calculation) or DO1 (during LM or SM operation).
3. Hazard MUX : Mux being controlled by the hazard block in MW stage.
4. Hazard\_MM : Checks the hazard during load instruction when the destination is R7. It suitably loads the PC value by controlling the hazard mux.

## Write Back

1. Flags\_user : The flags which are visible to user.
2. WB\_mux : It decides the input to be written in the register file. The inputs according to the instructions are
   1. ALU output : During arithmetic instructions.
   2. LS\_PC : LHI instruction.
   3. SE : LLI instruction.
   4. PC+1 : JAL, JLR instruction.
   5. Mem\_out : LW, LM instruction.
3. R7\_mux : It decides the input to the R7 in the register file. The inputs according to the instructions are
   1. DO1 : During JLR instruction.
   2. PC+1 : Normal Program Flow.
   3. LS\_PC : For BEQ instruction when the branch is taken.
4. Conditional and Hazard Control : Takes care of the following cases
   1. Conditional arithmetic instruction is not taken and dependency present in previous pipeline registers.
   2. Conditional arithmetic instruction when the destination is R7.
   3. JLR instruction when the destination is R7.
   4. JAL instruction when the destination is R7.
   5. Check for BEQ instruction if the branch is taken or not.
   6. Flushes the pipeline if any of the above condition is true.
5. Hazard Mux : Controlled by the above logic block. Decides the input to PC register.
   1. (SE+PC) when branch is taken in BEQ.
   2. PC+1 when JLR and JAL hazard is seen.
   3. Otherwise the default value coming at 0 input.