Basic Operation

1. Instruction Fetch
2. Decode
3. Operand Address Computation
4. Operand Fetch
5. Execution
6. Write-back of the result

Instruct. Address Computation
Instruction Formats (example):

Load/Store:

LD  RT  RB  DISPL

Operation

ADD  RT  RS1  RS2

Branch

BR  Address
Register - Transfer Operations: Addition Operation ADD

1. MABus <- IAR, R=1
2. MDBus <- Data (instruction)
3. IR <- Data
4. Decode Instruction (set the control lines to ADD)
5. SRA <- RS1, SRB <- RS2
6. SRC <- Result (SRA + SRB)
7. RT <- SRC
8. IAR <- IAR + 4
9. Check for Interrupts

ADD RT RS1 RS2

RT <= RS1 + RS2
LD : Load Instruction:

<table>
<thead>
<tr>
<th>LD</th>
<th>R3</th>
<th>Address</th>
</tr>
</thead>
</table>

R3 ← M[Address]

Major Cycles:

1. IF : Instruction Fetch
2. ID : Instruction Decode
3. OAG : Operand Address Generation
4. OF : Operand Fetch
5. EXEC : Execution (arithmetic / logical)
6. WB : Write Back (writing of the result)

IF: Instruction Fetch Cycle is the same for all instructions.

We need to check for the Interrupt and determine how are we going to handle Interrupt.
Register - Transfer Operations: Load Operation LD

1. MABus <- IAR, r = 1
2. MDBus <- Data (instruction)
3. IR <- Data
4. Decode Instruction
5. SRA <- RB, SRB <- Displacement
6. SRC <- Address
7. MAR <- SRC
8. MABuss <- MAR
9. MDBus <- Data
10. MDR <- MDBus
11. RT <- MDR
12. IAR <- IAR + 4
13. Check for INT
BR (C) Branch Instruction:

<table>
<thead>
<tr>
<th>BR</th>
<th>CC</th>
<th>Address</th>
</tr>
</thead>
</table>

**Major Cycles:**

1. IF: Instruction Fetch
2. ID: Instruction Decode
3. AG: Address Generation
4. Testing of the Condition CC
5. Writing into the IAR
1. MA_Bus ← IAR

2. R/W' ← 1, Enable IR to read data from the Memory Bus

3. Increment the IAR for the length of the currently decoded instruction:
   \[ \text{IAR} \leftarrow \text{IAR} + \text{IL} \] (IAR is now pointing to the next instruction)

4. Decide about interrupt
Register - Transfer Operations: Branch Operation BRC

1. MABus <- IAR
2. MDBus <- Data (instruction)
3. IR <- Data
4. Decode Instruction
5. Examine CC: if CC=1
   then:                     else:
6. IAR <- Address          6a. IAR <- IAR + 4
7. Check for INT