R TYPE - > instructions_test_1.hex

```
0x00418133
0x004202b3
0x40d605b3
0x005261b3
0x01e370b3
0x40b58533
0x00f6f533
0x009403b3
0x01498933
0x00e6e633
0x413908b3
0x01eeee33
0x01eefe33
0x00c5f533
0x000000b3
0x00000133
```

1. add x2, x3, x4
2. add x5, x4, x4
3. sub x11, x12, x13
4. or x3, x4, x5
5. and x1, x6, x30
6. sub x10, x11, x11
7. and x10, x13, x15
8. add x7, x8, x9
9. add x18, x19, x20
10. or x12, x13, x14
11. sub x17, x18, x19
12. or x28, x29, x30
13. and x28, x29, x30
14. and x10, x11, x12
15. add x1, x0, x0
16. add x2, x0, x0

BRANCH -> test_2.hex

```
0x00418133
0x004202b3
0x00a00513
0x00a00593
0x40310233
0x00b50463
0x01e370b3
0x40b58533
0x00f6f533
0x009403b3
0x01498933
0x00e6e633
0x00418133
0x004202b3
0x00a00513
0x00a00593
```

1. add x2, x3, x4
2. add x5,x4,x4
3. addi x10, x0, 10
4. addi x11, x0, 10
5. sub x4, x2, x3
6. beq x10, x11, 8 (12 pc)
7. and x1, x6, x30
8. sub x10, x11, x11
9. and x10, x13, x15
10. add x7, x8, x9
11. add x18, x19, x20
12. or x12, x13, x14
13. add x2, x3, x4
14. add x5,x4,x4
15. addi x10, x0, 10
16. addi x11, x0, 10

LOAD / STORE /ARITHMETIC -> instructions_test_3.hex

```
0x00803283
0x01003303
0x006283b3
0x00743423
0x00843583
0x00418133
0x004202b3
0x40d605b3
0x005261b3
0x01e370b3
0x40b58533
0x00f6f533
0x009403b3
0x01498933
0x00e6e633
0x413908b3
```

1. ld x5, 8(x0)
2. ld x6, 16(x0)
3. add x7, x5, x6
4. sd x7, 8(x8) stores to mem[2]
5. Ld x11,8(x8)
6. add x2, x3, x4
7. add x5, x4, x4
8. sub x11, x12, x13
9. or x3, x4, x5
10. and x1, x6, x30
11. sub x10, x11, x11
12. and x10, x13, x15
13. add x7, x8, x9
14. add x18, x19, x20
15. or x12, x13, x14
16. sub x17, x18, x19

1. **For control hazard**
   Same as seq_branch check

   1.0x00418133
2. 0x004202b3
3. 0x00a00513
4. 0x00a00593
5. 0x40310233
6. 0x00b50463
7. 0x01e370b3
8. 0x40b58533
9. 0x00f6f533
10. 0x00940   3b3
11. 0x01498933
12. 0x00e6e633
13. 0x00418133
14. 0x004202b3
15. 0x00a00513
16. 0x00a00593

1.add x2, x3, x4
2. add x5,x4,x4
3. addi x10, x0, 10
4. addi x11, x0, 10
5. sub x4, x2, x3
6. beq x10, x11, 8 (9th pc)
7. and x1, x6, x30
8. sub x10, x11, x11
9. and x10, x13, x15
10. add x7, x8, x9
11. add x18, x19, x20
12. or x12, x13, x14
13. add x2, x3, x4
14. add x5,x4,x4
15. addi x10, x0, 10
16. addi x11, x0, 10

Including load store beq

1. 0x00803103
2. 0x00803203
3. 0x00410463
4. 0x00418133
5. 0x004202b3
6. 0x40d605b3
7. 0x005261b3
8. 0x01e370b3
9. 0x40b58533
10. 0x00f6f533
11. 0x009403b3

1. Ld x2,8(x0)
2. Ld x4,8(x0)
3. Beq x2,x4,8 //jump to 6th instruction
4. add x2, x3, x4 x2=7
5. add x5, x4, x4 x5=8
6. sub x11, x12, x13 x11=-1
7. or x3, x4, x5 x3=5
8. and x1, x6, x30 x1=36
9. sub x10, x11, x11 x10=22
10. and x10, x13, x15 x10=27….
11. add x7, x8, x9

THIS HEX FILE IS ONE INDEX, SO PC = 24/4=6, U GO CHECK 7TH INSTRUCTION