RISC-V with Gem5

Building Gem5 for RISCV architechure

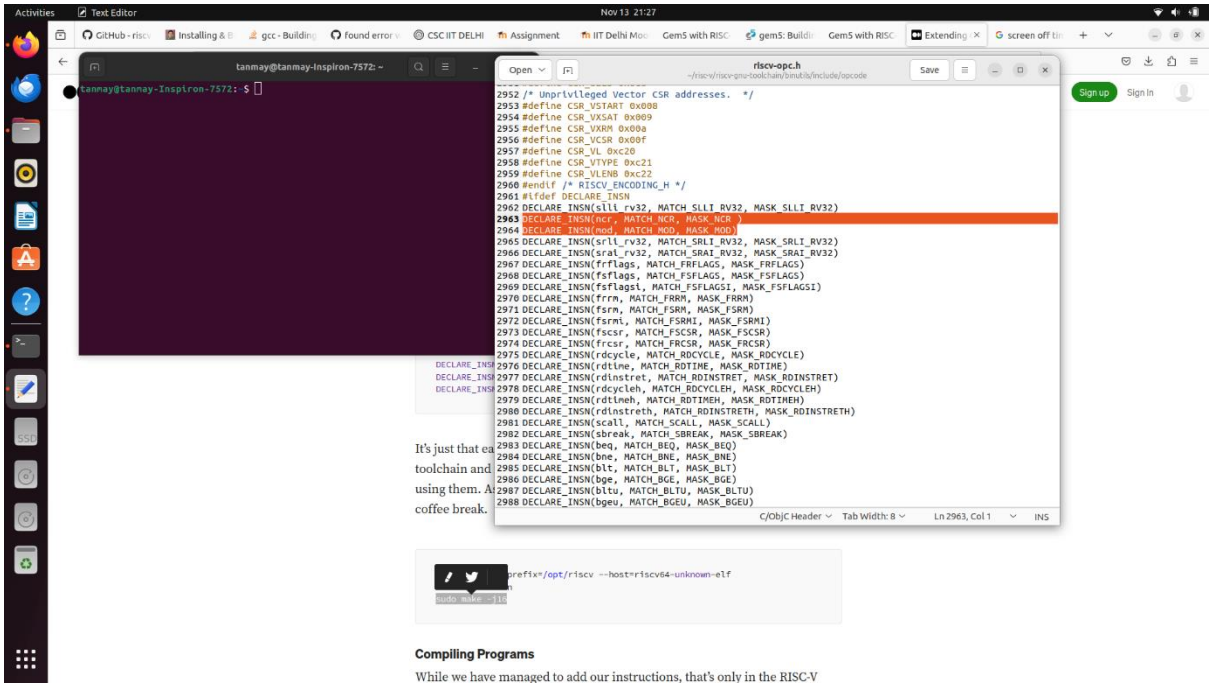
```
1    git clone https: //github.com/gem5/gem5
2    scons buil d /RISCV/gem5.opt -j 40
```

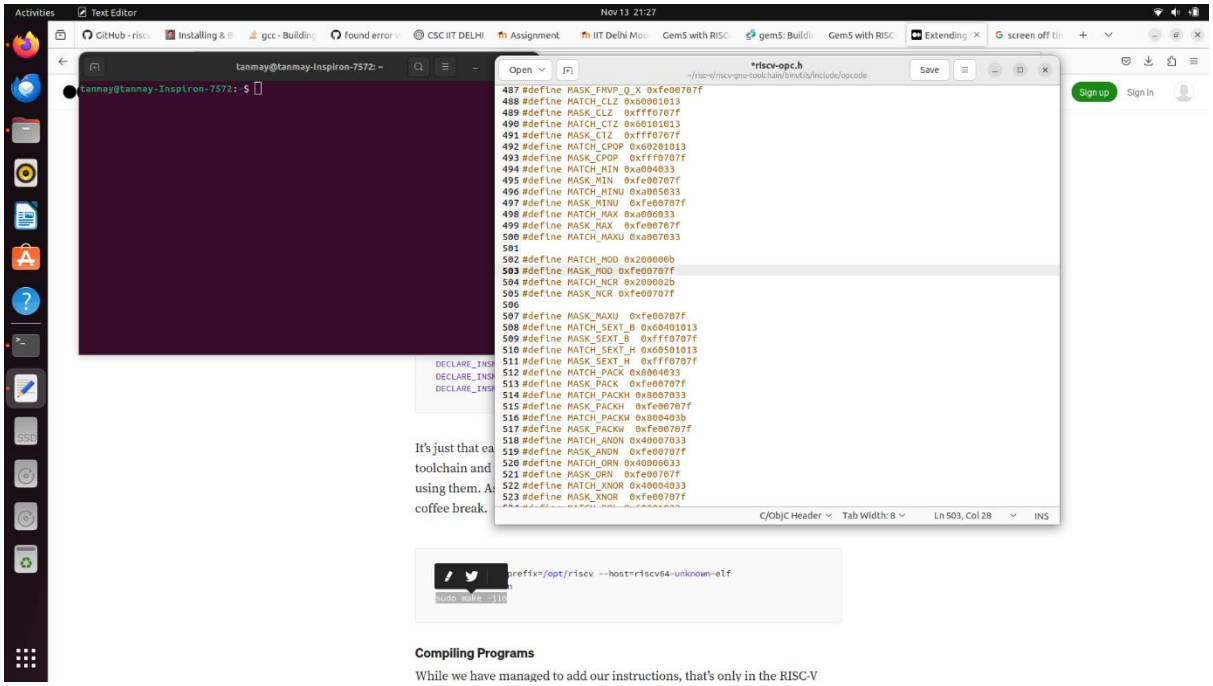
Build riscv-gnu-toolchain

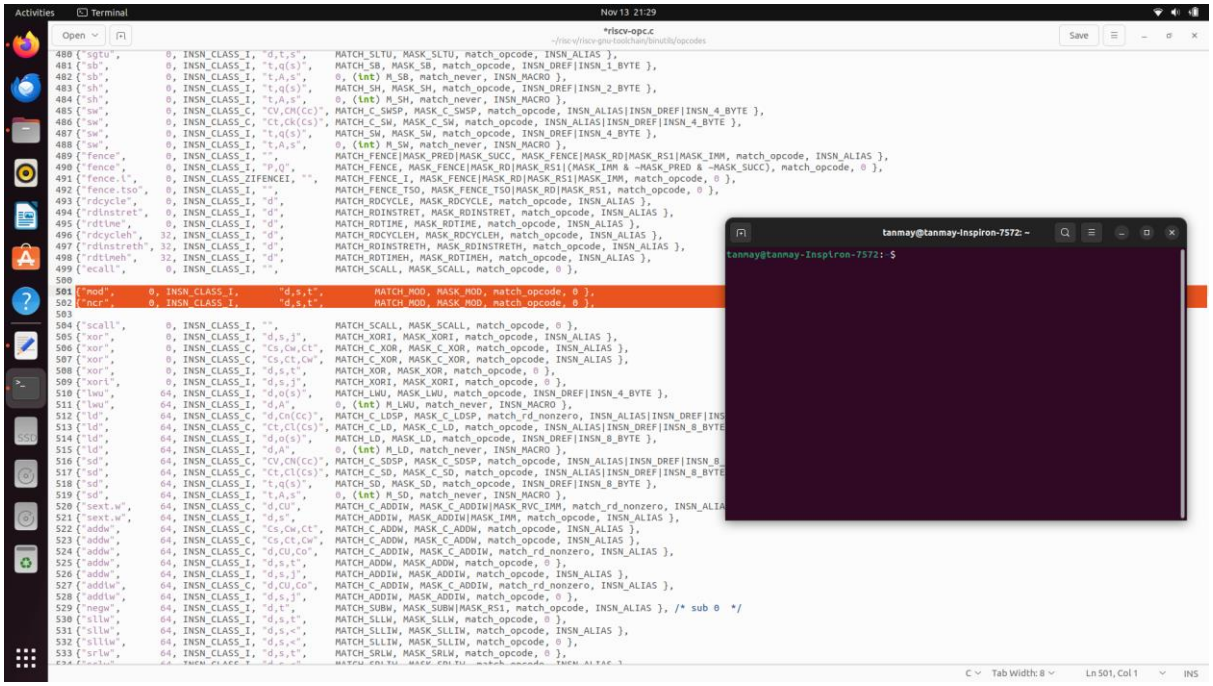
1. Downloaded the prerequisites
2. Cloned riscv-gnu-toolchain
3. Added the installation path to PATH
4. Build the toolchain

```
1    $ git clone https://github.com/riscv/riscv-gnu-toolchain
2    $ ./configure --prefix=/opt/riscv
3    $ sudo make linux
```

Adding Custom Instructions

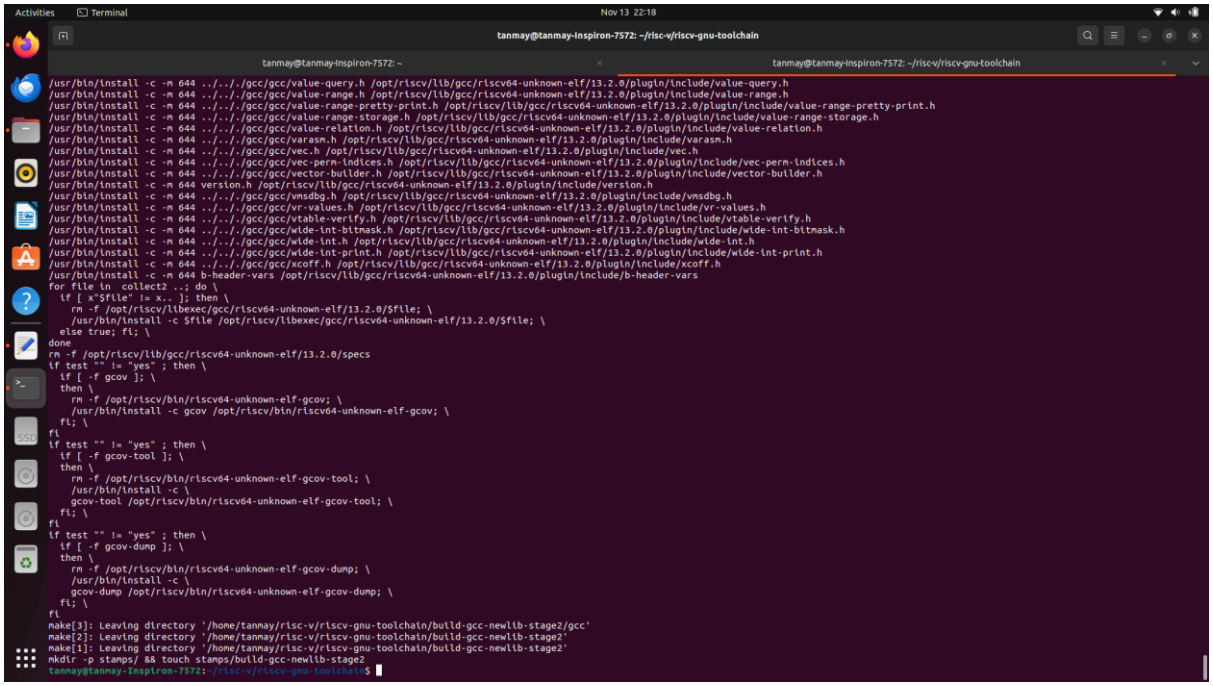






Rebuilt the toolchain

```
./configure --prefix=/opt/riscv --host=riscv64-unknown-elf
sudo make clean
sudo make -j16
```



Adding logic for nCr in decoder.isa

Location for decoder.isa is src/arch/{isa}/decoder.isa

```
0x07: decode FUNCT3 {
    format ROp {
        0x0: decode FUNCT7 {
            0x1: ncr({{
                uint64_t n = Rs1_sd;
                uint64_t r = Rs2_sd;
                r = r > (n-r) ? r : (n-r);
                uint64_t c = 1, fact = 1;
                for (uint64_t i = n; i > r; i--) {
                c = c * i;
                fact = fact * (n - i + 1);
                }
                Rd = c / fact;
            }});
        }
    }
}

```

Program-

```c
#include <stdint.h>
#include <stdio.h>

int main(void) {
    int64_t a = 3,b=5, n=7, sum=0;
    for (uint32_t i = 0; i <= n; i++) {
        uint32_t cn=n, cr=r;
        cr = cr > (cn-cr) ? cr : (cn-cr);
        uint64_t c = 1, fact = 1;
        for (uint64_t j = cn; j > cr; j--) {
            c = c * j;
            fact = fact * (n - j + 1);
        }
        int64_t comb= c/fact

        int powa = 1;
        for (uint32_t j = 1; j <= i; j++)
            y *= a;

        int powb = 1;
        for (uint32_t j = 1; j <= i; j++)
            y *= b;

        int64_t result = comb * powa * powb;
        sum += result;
        printf("%ld\t", result);
    }
    printf("\n%d\n", sum);

    return 0;
}
```

Issues: Getting errors while simulating the code in gem5