Student ID : 111062307,　　Name : 陳大佑

```
__data __at (0x30) char saved_SP[MAXTHREADS];
__data __at (0x34) ThreadID cur_ID;
__data __at (0x35) char bitmap_ID;
__data __at (0x36) char temp_SP;        // temporary save the SP
__data __at (0x37) ThreadID new_ID;
```

```
__data __at (0x38) char sharedBuffer;
__data __at (0x39) char bufferFull;     // 0: empty, 1 : full
__data __at (0x3A) char nextChar;
```

Set some parameters on manually allocated memory.

```
void Producer(void) {
    nextChar = 'A'; //
    while (1) {
        if (bufferFull) {
            ThreadYield(); // wait for buffer
            continue;
        }
        sharedBuffer = nextChar; // write for buffer
        bufferFull = 1; // mark buffer is full
        ThreadYield(); // switch
        nextChar = (nextChar - 'A' + 1) % 26 + 'A';
    }
}
```

Since we want to print ABC...YZABC for loop so I use nextChar = (...) % 26 + 'A', and when bufferFull = 1, it means we have value now, and we can switch to consumer to print it.

```
void Consumer(void) {
    initialize();
    while (1) {
        if (!bufferFull) {
            ThreadYield(); // wait for buffer
            continue;
        }
        SBUF = sharedBuffer; // write the content of buffer
        while (!TI); // wait trans
        TI = 0; // mark done
        bufferFull = 0; // mark buffer is empty
        ThreadYield(); // switch
    }
}
```

First we initialize it, and check whether buffer is full or not, then print it or switch it.

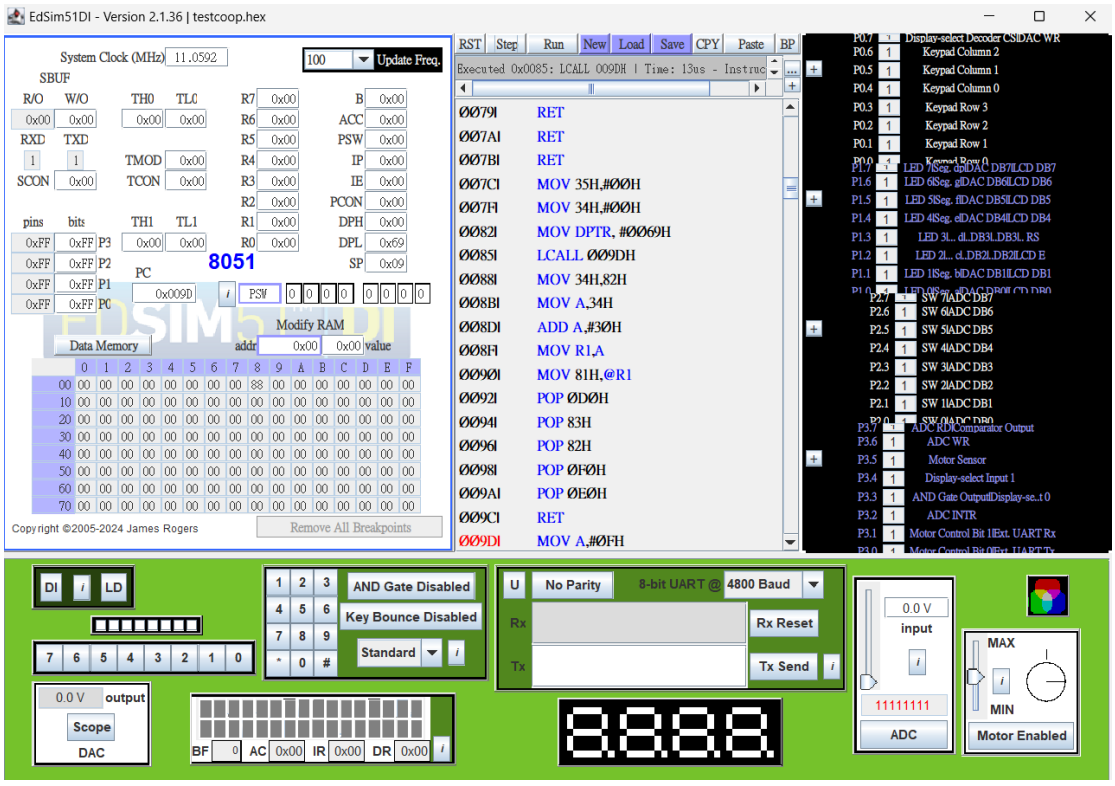
```
void initialize(){
    TMOD = 0x20;
    SCON = 0x50;
    TH1 = (char) -6;
    TR1 = 1;
    TI = 1;
}
```

Screenshots for compliation.

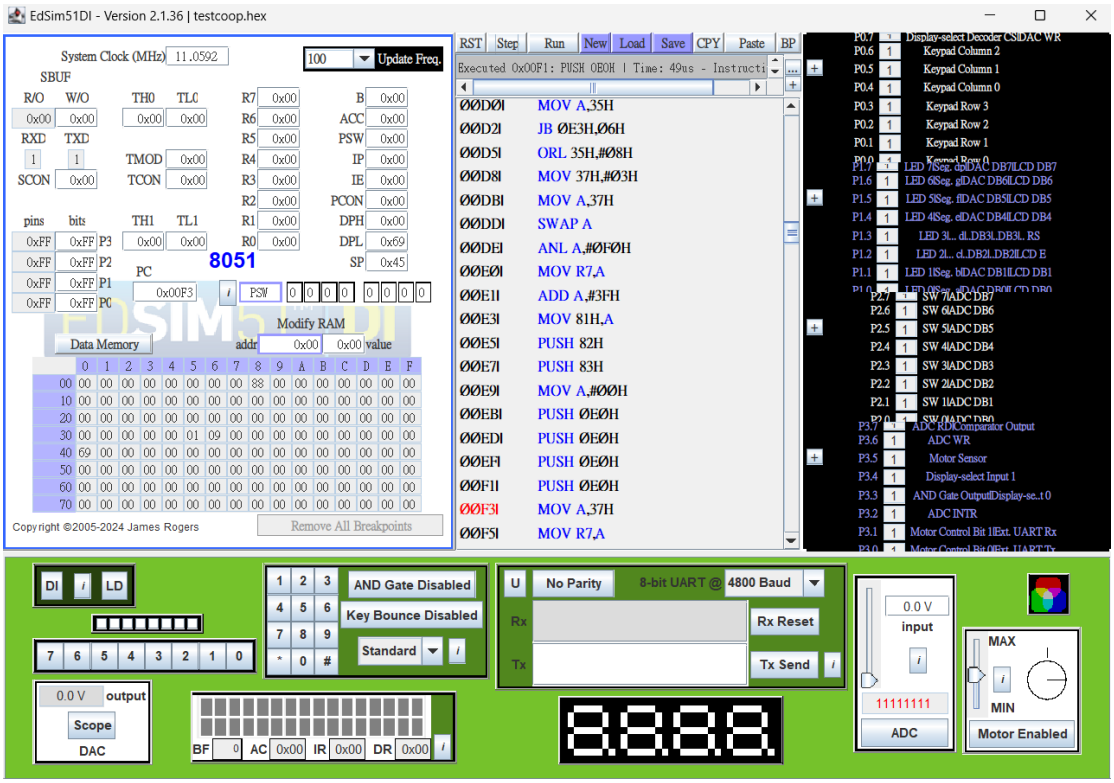
```
dylan@LAPTOP-S5FOLV5V ~/os/ppc1
$ make clean
rm *.hex *.ihx *.lnk *.lst *.map *.mem *.rel *.rst *.sym *.asm *.lk
rm: cannot remove '*.ihx': No such file or directory
rm: cannot remove '*.lnk': No such file or directory
make: *** [clean] Error 1

dylan@LAPTOP-S5FOLV5V ~/os/ppc1
$ make
sdcc -c  testcoop.c
sdcc -c  cooperative.c
cooperative.c:154: warning 85: in function ThreadCreate unreferenced function argument : 'fp'
sdcc  -o testcoop.hex testcoop.rel cooperative.rel
```

Screenshots and explanation:

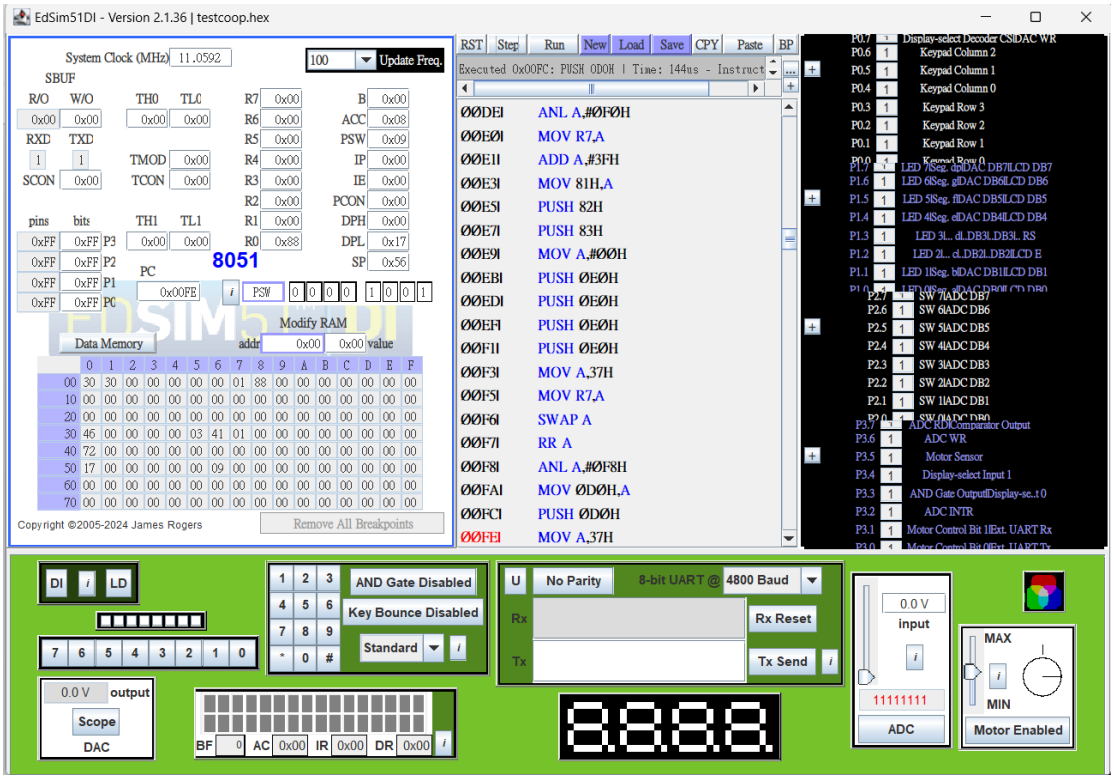
Before the main thread was created, since by default the SP = 0x07, the 0x08 was numbered by first instruction(bootstrap).



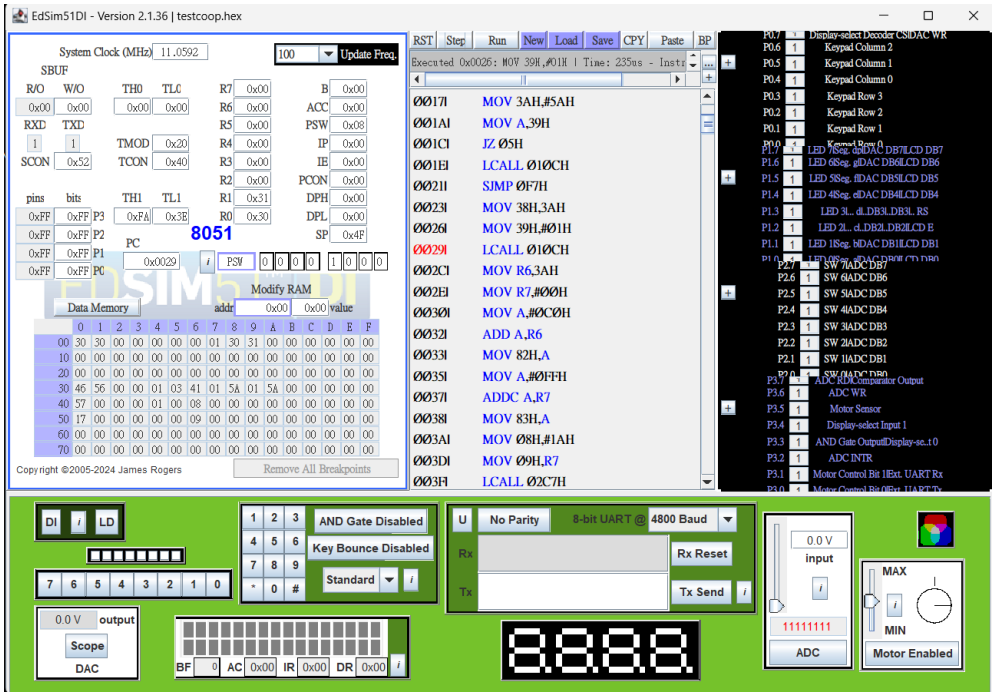
After main thread had been created, the address of 40H was stored for main's address, and 42H, 43H, 44H, 45H, 46H are ACC, B, DPL, DPH, PSW, respectively.



And after producer was created, the address of the 50H was stored for producer's address. and 52H~56H are ACC, B, DPL, DPH, PSW, respectively.



We can know the producer is running, since cur_ID on address(34H) is 01 which means producer and compare to previous picture we can find the addresses of(38H, 39H, 3AH) are replaced by other number(char'A').



We can know the consumer is running, since cur_ID on address(34H) is 00 which means it switches. And the address(39H) is changed from 1 to 0.