ICS Lab2 Report

PB20000096 潘廷岳

Part1.version L

This lab is easy to deal with.We just need to initialize the F0,F1 and F2,then calculating from F[3] to F[n] according to the recursion formula,and put the F[n] to the R7,then the process is finished.I use 23 lines to finish it.

And more,don't forget to MOD the MOD number!

The follows is the solution of the problem,expressed by assembly language.

```
.ORIG x3000
ADD R1, R1, #1 ;
ADD R2, R2, #1 ;
LD R6, MOD ;/Memory the MOD number
ADD R7, R7, #1 ;
ADD R0, R0, #-1;
BRz LOOP ;
ADD R7, R7, #1 ;
ADD R0, R0, #-1;
BRz LOOP ;
LOOP1 ADD R3, R2, #0;
ADD R2, R1, #0;
ADD R1, R7, #0;
ADD R5, R3, R3;//R3=R3*2
ADD R7, R1, R5;
AND R7, R7, R6;//R7=R7%(1<<10);
ADD R0, R0, #-1;
BRp LOOP1;
LOOP HALT ;
MOD .FILL x03FF;
F(a) .FILL x0398;//F[20]%1024=(920)D
F(b) .FILL x0001;//F[0]%1024=(1)D
F(c) .FILL x0001;//F[0]%1024=(1)D
F(d) .FILL x0102;//F[96]%1024=(258)D
.END
```

Part2.Correctness Validation

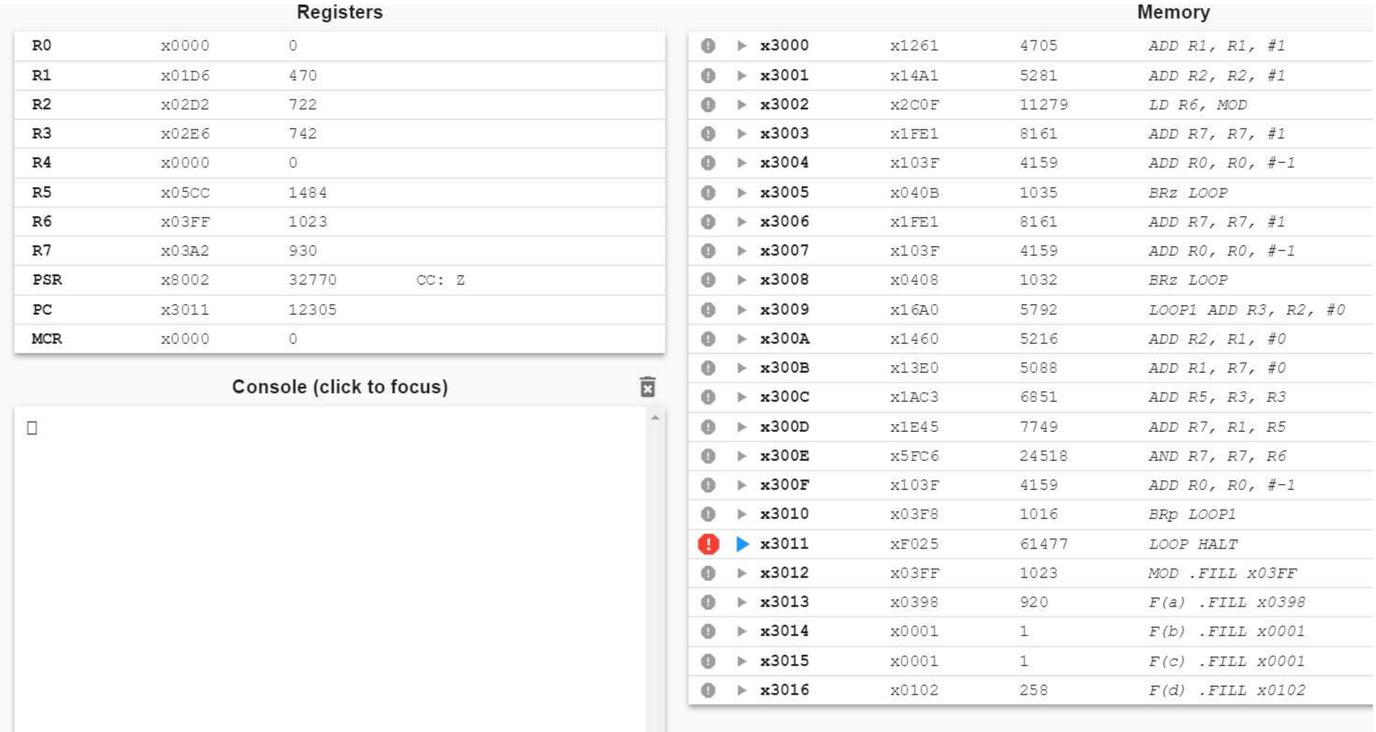
To valid the the correctness of this assembly program,I wrote a program based on C programming language as follows:

```
int main()
{
    int Mod=1024;
    int n;
    while(1)
    {
        int F0=1,F1=1,F2=2,Fn=0;
        cin>>n;
        if(n==-1) break;
        if(n==1) cout<<"1";
        else if(n==2) cout<<"2";
        else{
            n-=2;
            while(n--)
            {
                Fn=(F2+2*F0)%Mod;
                F0=F1;F1=F2;F2=Fn;
            }
        }
        cout<<Fn<<endl;
    }
    return 0;
}
```

After several tests，the answers are all correct,that's perfect!

Part3.Photo Display

n=20:



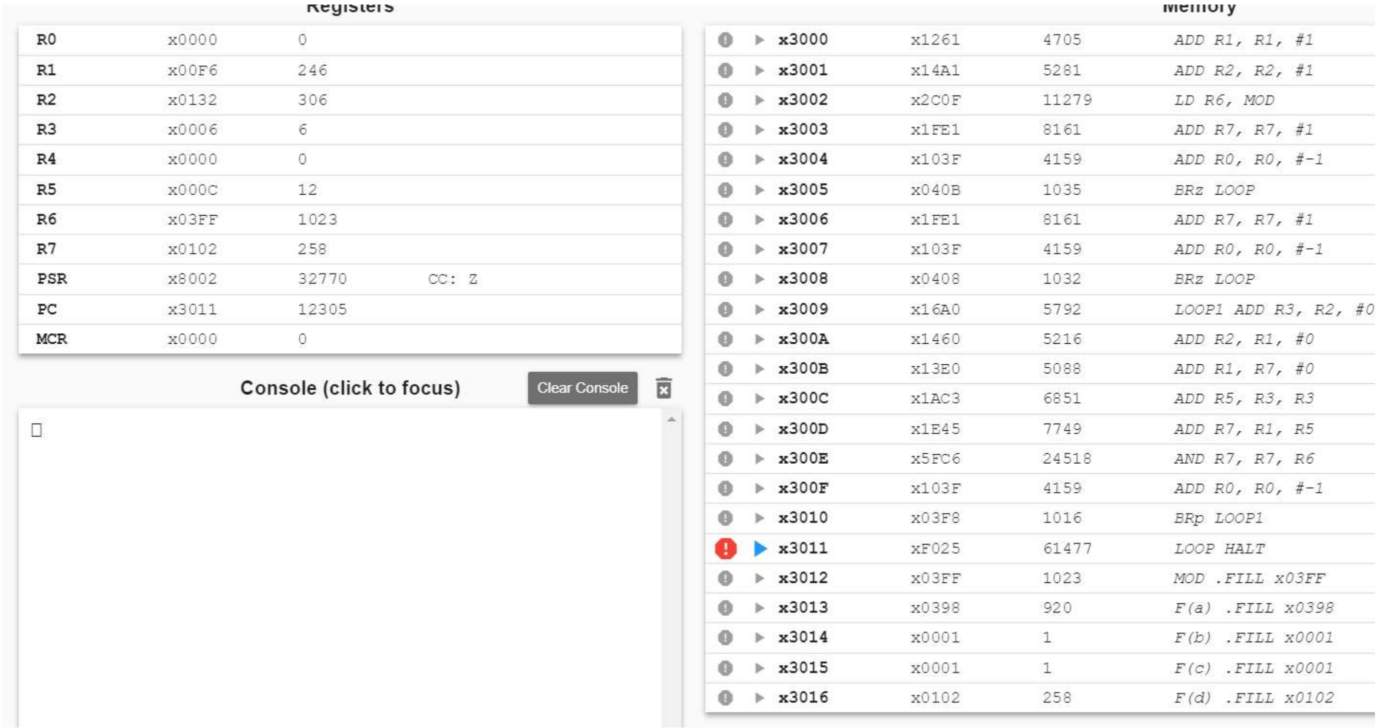
Registers

R0	x0000	0	
R1	x01D6	470	
R2	x02D2	722	
R3	x02E6	742	
R4	x0000	0	
R5	x05CC	1484	
R6	x03FF	1023	
R7	x03A2	930	
PSR	x8002	32770	CC: Z
PC	x3011	12305	
MCR	x0000	0	

Memory

x3000	x1261	4705	ADD R1, R1, #1
x3001	x14A1	5281	ADD R2, R2, #1
x3002	x2C0F	11279	LD R6, MOD
x3003	x1FE1	8161	ADD R7, R7, #1
x3004	x103F	4159	ADD R0, R0, #-1
x3005	x040B	1035	BRz LOOP
x3006	x1FE1	8161	ADD R7, R7, #1
x3007	x103F	4159	ADD R0, R0, #-1
x3008	x0408	1032	BRz LOOP
x3009	x16A0	5792	LOOP1 ADD R3, R2, #0
x300A	x1460	5216	ADD R2, R1, #0
x300B	x13E0	5088	ADD R1, R7, #0
x300C	x1AC3	6851	ADD R5, R3, R3
x300D	x1E45	7749	ADD R7, R1, R5
x300E	x5FC6	24518	AND R7, R7, R6
x300F	x103F	4159	ADD R0, R0, #-1
x3010	x03F8	1016	BRp LOOP1
x3011	xF025	61477	LOOP HALT
x3012	x03FF	1023	MOD .FILL x03FF
x3013	x0398	920	F(a) .FILL x0398
x3014	x0001	1	F(b) .FILL x0001
x3015	x0001	1	F(c) .FILL x0001
x3016	x0102	258	F(d) .FILL x0102

n=96:



Registers

R0	x0000	0	
R1	x00F6	246	
R2	x0132	306	
R3	x0006	6	
R4	x0000	0	
R5	x000C	12	
R6	x03FF	1023	
R7	x0102	258	
PSR	x8002	32770	CC: Z
PC	x3011	12305	
MCR	x0000	0	

Console (click to focus)

Memory

x3000	x1261	4705	ADD R1, R1, #1
x3001	x14A1	5281	ADD R2, R2, #1
x3002	x2C0F	11279	LD R6, MOD
x3003	x1FE1	8161	ADD R7, R7, #1
x3004	x103F	4159	ADD R0, R0, #-1
x3005	x040B	1035	BRz LOOP
x3006	x1FE1	8161	ADD R7, R7, #1
x3007	x103F	4159	ADD R0, R0, #-1
x3008	x0408	1032	BRz LOOP
x3009	x16A0	5792	LOOP1 ADD R3, R2, #0
x300A	x1460	5216	ADD R2, R1, #0
x300B	x13E0	5088	ADD R1, R7, #0
x300C	x1AC3	6851	ADD R5, R3, R3
x300D	x1E45	7749	ADD R7, R1, R5
x300E	x5FC6	24518	AND R7, R7, R6
x300F	x103F	4159	ADD R0, R0, #-1
x3010	x03F8	1016	BRp LOOP1
x3011	xF025	61477	LOOP HALT
x3012	x03FF	1023	MOD .FILL x03FF
x3013	x0398	920	F(a) .FILL x0398
x3014	x0001	1	F(b) .FILL x0001
x3015	x0001	1	F(c) .FILL x0001
x3016	x0102	258	F(d) .FILL x0102