TI DSP, MCU, Xilinx Zynq FPGA Based Programming Expert Program

Instructor – Innova Lee (Sanghoon Lee)
gcccompil3r@gmail.com
Student – Howard Kim (Hyungjoo Kim)
mihaelkel@naver.com

Mov r0, r1, N

```
1 #include <stdio.h>
2
3 void show_reg(unsigned int reg){
4    int i;
5
6    for(i=31;i>=0;)
7         printf("%d",(reg>>i--)&1);
8    printf("\n");
9
10 }
11
12 int main(void){
13
14    register unsigned int r0 asm("r0") = 0;
15    register unsigned int r1 asm("r1") = 0;
16    register unsigned int r2 asm("r2") = 0;
17    register unsigned int r3 asm("r3") = 0;
18    register unsigned int r4 asm("r4") = 0;
19    register unsigned int r5 asm("r5") = 0;
20
21    asm volatile("mov r0,#0xff,8");
22
23    printf("r0 = 0x%x\n",r0);
24    return 0;
25 }
```

```
howi@ubuntu:~/HomeworkBackup/45th$ arm-linux-gnueabi-gcc -g asm11.c
howi@ubuntu:~/HomeworkBackup/45th$ qemu-arm-static -L /usr/arm-linux-gnueabi ./a.out
r0 = 0xff000000
```

lsl option

```
#include <stdio.h>

include <stdio.h>

void show_reg(unsigned int reg){
    int i;

for(i=31;i>=0;)
    printf("%d",(reg>>i--)&1);
    printf("\n");

printf("\n");

register unsigned int r0 asm("r0") = 0;
    register unsigned int r1 asm("r1") = 0;
    register unsigned int r2 asm("r2") = 0;
    register unsigned int r3 asm("r3") = 0;
    register unsigned int r4 asm("r4") = 0;
    register unsigned int r5 asm("r5") = 0;
    register unsigned int r5 asm("r5") = 0;
    register unsigned int r5 asm("r5") = 0;
    register unsigned int r5 asm("r5");
    asm volatile("mov r1,#7");
    asm volatile("mov r2,#3");
    asm volatile("add r0,r1,r2,lsl #7");

printf("r0 = %d\n",r0);
    return 0;
```

howi@ubuntu:~/HomeworkBackup/45th\$ qemu-arm-static -L /usr/arm-linux-gnueabi ./a.out r0 = 391

```
asm15.c(-/HomeworkBackup/45th) - VIM

#include <stdio.h>

void show_reg(unsigned int reg){
    int i;

for(i=31;i>=0;)
    printf("%d",(reg>>i--)&1);
    printf("\n");

printf("\n");

register unsigned int r0 asm("r0") = 0;
    register unsigned int r1 asm("r1") = 0;
    register unsigned int r1 asm("r2") = 0;
    register unsigned int r3 asm("r2") = 0;
    register unsigned int r4 asm("r4") = 0;
    register unsigned int r5 asm("r5") = 0;
    register unsigned int r6 asm("r5") = 0;
    register unsigned int r7 asm("r5") = 0;
    register
```

howi@ubuntu:~/HomeworkBackup/45th\$ qemu-arm-static -L /usr/arm-linux-gnueabi ./a.out r0 = 0xa

asr option

```
1 #include <stdio.h>
   void show reg(unsigned int reg){
 4
          int i;
 5
 б
          for(i=31;i>=0;)
                printf("%d",(reg>>i--)&1);
 7
          printf("\n");
 8
 9
10 }
11
12 int main(void){
13
         register unsigned int r0 asm("r0") = 1;
register unsigned int r1 asm("r1") = 0;
register unsigned int r2 asm("r2") = 0;
register unsigned int r3 asm("r3") = 0;
14
15
16
17
18
          register unsigned int r4 asm("r4") = 0;
19
          register unsigned int r5 asm("r5") = 0;
20
         asm volatile("mov r1,#32");
asm volatile("add r0,r1,asr #2");
21
22
23
24
          printf("r0 = 0x%x\n",r0);
25
          return 0:
26
```

```
howi@ubuntu:~/HomeworkBackup/45th$ arm-linux-gnueabi-gcc -g asm16.c
howi@ubuntu:~/HomeworkBackup/45th$ qemu-arm-static -L /usr/arm-linux-gnueabi ./a.out
r0 = 0x9
```

mrs(Mov Register Statusregister)

```
#include <stdio.h>

void show_reg(unsigned int reg){
    int i;

for(i=31;i>=0;)
    printf("%d",(reg>>i--)&1);

printf("\n");

printf("\n");

register unsigned int r0 asm("r0") = 0;
    register unsigned int r1 asm("r1") = 0;
    register unsigned int r2 asm("r2") = 0;
    register unsigned int r3 asm("r2") = 0;
    register unsigned int r3 asm("r3") = 0;
    register unsigned int r3 asm("r5") = 0;
    register unsigned int r5 asm("r5") = 0;
    register unsigned int r5 asm("r5") = 0;
    sam volatile("mov r1,#32");
    asm volatile("mov r1,#32");
    asm volatile("mrs r0,cpsr");

show_reg(r0);
    return 0;

return 0;
```

```
mul a, b, c
```

```
1 #include <stdio.h>
 3 void show_reg(unsigned int reg){
 4
          int i;
 5
 б
          for(i=31;i>=0;)
 7
                printf("%d",(reg>>i--)&1);
 8
          printf("\n");
 9
10 }
11
12 int main(void){
13
          register unsigned int r0 asm("r0") = 0;
register unsigned int r1 asm("r1") = 0;
register unsigned int r2 asm("r2") = 0;
register unsigned int r3 asm("r3") = 0;
register unsigned int r4 asm("r4") = 0;
14
15
16
17
18
19
          register unsigned int r5 asm("r5") = 0;
20
21
          asm volatile("mov r2,#3");
22
          asm volatile("mov r3,#7");
          asm volatile("mul r1,r2,r3");
23
24
25
          printf("r1=%d\n",r1);
26
    II
          show reg(r0);
27
          return 0;
28 }
```

howi@ubuntu:~/HomeworkBackup/45th\$ arm-linux-gnueabi-gcc -g asm18.c howi@ubuntu:~/HomeworkBackup/45th\$ qemu-arm-static -L /usr/arm-linux-gnueabi ./a.out r1=21 mla a, b, c, d (MuLtiply-Accumulate)

```
#include <stdio.h>
 3
    void show_reg(unsigned int reg){
           for(i=31;i>=0;)
    printf("%d",(reg>>i--)&1);
printf("\n");
 8
 9
10 }
11
12 int main(void){
14
           register unsigned int r0 asm("r0") = 0;
           register unsigned int r1 asm("r1") = 0;
register unsigned int r2 asm("r2") = 0;
register unsigned int r3 asm("r3") = 0;
register unsigned int r4 asm("r4") = 0;
15
16
18
19
20
21
22
           register unsigned int r5 asm("r5") = 0;
           asm volatile("mov r2,#3");
asm volatile("mov r3,#7");
asm volatile("mov r4,#33");
23
24
25
           asm volatile("mla r1,r2,r3,r4");
26
           printf("r1=%d\n",r1);
          show_reg(r0);
28
           return 0;
29
```

howi@ubuntu:~/HomeworkBackup/45th\$ arm-linux-gnueabi-gcc -g asm19.c howi@ubuntu:~/HomeworkBackup/45th\$ qemu-arm-static -L /usr/arm-linux-gnueabi ./a.out r1=54

umull a, b, c, d

```
1 #include <stdio.h>
  3 void show_reg(unsigned int reg){
  4
5
              int i;
  6
7
              for(i=31;i>=0;)
                     printf("%d",(reg>>i--)&1);
  8
              printf("\n");
  9
     int main(void){
12
13
14
15
16
              register unsigned int r0 \text{ asm}("r0") = 0;
             register unsigned int r0 asm("r0") = 0;
register unsigned int r1 asm("r1") = 0;
register unsigned int r2 asm("r2") = 0;
register unsigned int r3 asm("r3") = 0;
register unsigned int r4 asm("r4") = 0;
register unsigned int r5 asm("r5") = 0;
17
18
19
20
21
22
23
             asm volatile("mov r2,#0x44,8");
asm volatile("mov r3,#0x200");
asm volatile("umull r0,r1,r2,r3");
24
25
              printf("r1r0=0x%x%08x\n",r1,r0);
26
              show_reg(r0);
27
              return 0;
28
```

howi@ubuntu:~/HomeworkBackup/45th\$ arm-linux-gnueabi-gcc -g asm20.c howi@ubuntu:~/HomeworkBackup/45th\$ qemu-arm-static -L /usr/arm-linux-gnueabi ./a.out r1r0=0x8800000000 umlal a, b, c, d

```
#include <stdio.h>
     void show_reg(unsigned int reg){
            int i;
           for(i=31;i>=0;)
    printf("%d",(reg>>i--)&1);
printf("\n");
 9
     int main(void){
12
13
           register unsigned int r0 asm("r0") = 0;
register unsigned int r1 asm("r1") = 0;
register unsigned int r2 asm("r2") = 0;
14
15
           register unsigned int r3 asm("r3") = 0;
16
17
18
           register unsigned int r4 asm("r4") = 0;
register unsigned int r5 asm("r5") = 0;
19
           asm volatile("mov r0,#0xf");
asm volatile("mov r1,#0x1");
20
21
           asm volatile("mov r2,#0x44, 8");
asm volatile("mov r3,#0x200");
asm volatile("umlal r0,r1,r2,r3");
22
23
24
26
           printf("r1r0=0x%x%08x\n",r1,r0);
27
28
           show_reg(r0);
29
            return 0:
30 }
```

howi@ubuntu:~/HomeworkBackup/45th\$ arm-linux-gnueabi-gcc -g asm21.c
howi@ubuntu:~/HomeworkBackup/45th\$ qemu-arm-static -L /usr/arm-linux-gnueabi ./a.out
r1r0=0x890000000f

ldr a, [b]

```
1 #include <stdio.h>
2 unsigned int arr[5] = {1, 2, 3, 4, 5};
3 void show_reg(unsigned int reg){
           int i;
 5
            for(i=31;i>=0;)
 б
                 printf("%d",(reg>>i--)&1);
 8
           printf("\n");
 9
11 int main(void)
13
           register unsigned int r0 \text{ asm}("r0") = 0;
           register unsigned int *r1 asm("r1") = NULL;
register unsigned int *r2 asm("r2") = NULL;
register unsigned int r3 asm("r3") = 0;
register unsigned int r4 asm("r4") = 0;
14
15
16
17
18
19
20
21
22
23
24
25
           register unsigned int r5 asm("r5") = 0;
           r1 = arr;
           asm volatile("mov r2,#0x4");
asm volatile("ldr r0, [r1, #0x4]");
           printf("r0=%u, r1=%u\n",r0, *r1);
26
27
           show reg(r0);
28
           return 0;
29 }
```

howi@ubuntu:~/HomeworkBackup/45th\$ arm-linux-gnueabi-gcc -g asm22.c
howi@ubuntu:~/HomeworkBackup/45th\$ qemu-arm-static -L /usr/arm-linux-gnueabi ./a.out
r0=2, r1=1

```
#include <stdio.h>
 2 char test[32] = "HelloARM";
3 void show_reg(unsigned int reg){
              for(i=31;i>=0;)
              printf("%d",(reg>>i--)&1);
printf("\n");
 8
 9
10
      int main(void){
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
              register unsigned int r0 asm("r0") = 0;
             register unsigned int ro asm( ro ) = 0;
register char *r1 asm("r1") = NULL;
register unsigned int *r2 asm("r2") = NULL;
register unsigned int r3 asm("r3") = 0;
register unsigned int r4 asm("r4") = 0;
register unsigned int r5 asm("r5") = 0;
              r1 = &test[5];
             asm volatile("mov r0, #97");
asm volatile("strb r0,[r1]");
              printf("test=%s\n",test);
            show_reg(r0);
28
              return 0;
29 }
```

howi@ubuntu:~/HomeworkBackup/45th\$ arm-linux-gnueabi-gcc -g asm24.c howi@ubuntu:~/HomeworkBackup/45th\$ qemu-arm-static -L /usr/arm-linux-gnueabi ./a.out test=HelloaRM

ldr a, [b, c]

```
#include <stdio.h;</pre>
  char test[32] = "HelloARM";
void show_reg(unsigned int reg){
  4
               int i:
               for(i=31;i>=0;)
    printf("%d",(reg>>i--)&1);
printf("\n");
  6
7
      int main(void){
12
13
14
              register char r0 asm("r0") = 0;
register char *r1 asm("r1") = NULL;
register unsigned int r2 asm("r2") = 0;
register unsigned int r3 asm("r3") = 0;
register unsigned int r4 asm("r4") = 0;
register unsigned int r5 asm("r5") = 0;
16
17
18
19
20
21
22
               r1 = test;
               asm volatile("mov r2, #0x5");
asm volatile("ldr r0,[r1,r2]");
23
24
25
26
               printf("test=%s, r1=%s, r0=%c\n",test, r1, r0);
              show_reg(r0);
28
29 }
```

howi@ubuntu:~/HomeworkBackup/45th\$ arm-linux-gnueabi-gcc -g asm25.c howi@ubuntu:~/HomeworkBackup/45th\$ qemu-arm-static -L /usr/arm-linux-gnueabi ./a.out test=HelloARM, r1=HelloARM, r0=A ldr a, [b], c

```
1 #include <stdio.h>
 2 unsigned int arr[5] = \{1, 2, 3, 4, 5\};
 3 void show_reg(unsigned int reg){
          int i;
 5
6
          for(i=31;i>=0;)
    printf("%d",(reg>>i--)&1);
 8
          printf("\n");
 9
10 }
11 int main(void){
12
         register unsigned int r0 asm("r0") = 0;
register unsigned int *r1 asm("r1") = NULL;
register unsigned int *r2 asm("r2") = NULL;
register unsigned int r3 asm("r3") = 0;
register unsigned int r4 asm("r4") = 0;
13
14
15
16
17
18
          register unsigned int r5 asm("r5") = 0;
19
20
          r1 = arr;
21
22
          asm volatile("mov r2, #0x10");
23
24
25
          asm volatile("ldr r0,[r1],r2");
          printf("r0=%u,r1 = %u\n",r0,*r1);
26
27 //
28
         show_reg(r0);
          return 0;
29 }
```

```
howi@ubuntu:~/HomeworkBackup/45th$ arm-linux-gnueabi-gcc -g asm26.c
howi@ubuntu:~/HomeworkBackup/45th$ qemu-arm-static -L /usr/arm-linux-gnueabi ./a.out
r0=1,r1 = 5
```