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

```c
#include <stdio.h>

void show_reg(unsigned int reg){
    int i;

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

}

int main(void){

    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;

    asm volatile("mov r0,#0xff,8");

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

```
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

```c
#include <stdio.h>

void show_reg(unsigned int reg){
    int i;

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

}

int main(void){

    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;

    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

```c
#include <stdio.h>

void show_reg(unsigned int reg){
    int i;

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

}

int main(void){

    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;

    asm volatile("mov r1,#2");
    asm volatile("add r0,r1,r1,lsl #2");

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

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

asr option

```c
#include <stdio.h>

void show_reg(unsigned int reg){
    int i;

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

}

int main(void){

    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;
    register unsigned int r4 asm("r4") = 0;
    register unsigned int r5 asm("r5") = 0;

    asm volatile("mov r1,#32");
    asm volatile("add r0,r1,asr #2");

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

```
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)

```c
#include <stdio.h>

void show_reg(unsigned int reg){
    int i;

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

}

int main(void){

    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;

    asm volatile("mov r1,#32");
    asm volatile("add r0,r1,asr #2");
    asm volatile("mrs r0,cpsr");

    show_reg(r0);
    return 0;
}
```

```
howi@ubuntu:~/HomeworkBackup/45th$ arm-linux-gnueabi-gcc -g asm17.c
howi@ubuntu:~/HomeworkBackup/45th$ qemu-arm-static -L /usr/arm-linux-gnueabi ./a.out
01100000000000000000000000010000
```

mul a, b, c

```c
#include <stdio.h>

void show_reg(unsigned int reg){
    int i;

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

}

int main(void){

    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;

    asm volatile("mov r2,#3");
    asm volatile("mov r3,#7");
    asm volatile("mul r1,r2,r3");

    printf("r1=%d\n",r1);
//  show_reg(r0);
    return 0;
}
```

```
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)

```c
#include <stdio.h>

void show_reg(unsigned int reg){
    int i;

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

}

int main(void){

    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;

    asm volatile("mov r2,#3");
    asm volatile("mov r3,#7");
    asm volatile("mov r4,#33");
    asm volatile("mla r1,r2,r3,r4");

    printf("r1=%d\n",r1);
//  show_reg(r0);
    return 0;
}
```

```
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

```c
#include <stdio.h>

void show_reg(unsigned int reg){
    int i;

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

}
int main(void){

    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;

    asm volatile("mov r2,#0x44,8");
    asm volatile("mov r3,#0x200");
    asm volatile("umull r0,r1,r2,r3");

    printf("r1r0=0x%x%08x\n",r1,r0);

//  show_reg(r0);
    return 0;
}
```

```
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

```c
#include <stdio.h>

void show_reg(unsigned int reg){
    int i;

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

}
int main(void){

    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;

    asm volatile("mov r0,#0xf");
    asm volatile("mov r1,#0x1");
    asm volatile("mov r2,#0x44, 8");
    asm volatile("mov r3,#0x200");
    asm volatile("umlal r0,r1,r2,r3");

    printf("r1r0=0x%x%08x\n",r1,r0);

//  show_reg(r0);
    return 0;
}
```

```
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]

```c
#include <stdio.h>
unsigned int arr[5] = {1, 2, 3, 4, 5};
void show_reg(unsigned int reg){
    int i;

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

}
int main(void){

    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;
    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);

//  show_reg(r0);
    return 0;
}
```

```
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
```

strb

```c
#include <stdio.h>
char test[32] = "HelloARM";
void show_reg(unsigned int reg){
    int i;

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

}
int main(void){

    register unsigned int r0 asm("r0") = 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);
    return 0;
}
```

```
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]

```c
#include <stdio.h>
char test[32] = "HelloARM";
void show_reg(unsigned int reg){
    int i;

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

}
int main(void){

    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;

    r1 = test;

    asm volatile("mov r2, #0x5");
    asm volatile("ldr r0,[r1,r2]");

    printf("test=%s, r1=%s, r0=%c\n",test, r1, r0);

//  show_reg(r0);
    return 0;
}
```

```
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

```c
#include <stdio.h>
unsigned int arr[5] = {1, 2, 3, 4, 5};
void show_reg(unsigned int reg){
    int i;

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

}
int main(void){

    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;
    register unsigned int r5 asm("r5") = 0;

    r1 = arr;

    asm volatile("mov r2, #0x10");
    asm volatile("ldr r0,[r1],r2");

    printf("r0=%u,r1 = %u\n",r0,*r1);

//  show_reg(r0);
    return 0;
}
```

```
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
```