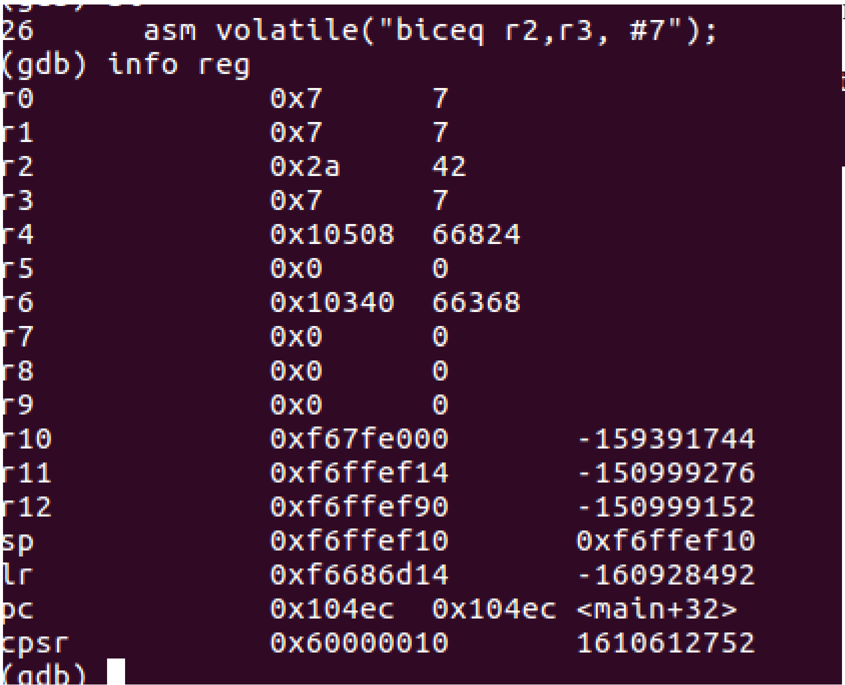
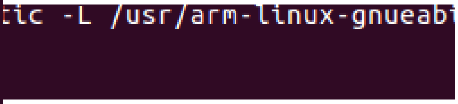
I DSP,Xilinx zynq FPGA,MCU 및 Xilinx zynq FPGA 프로그래밍 전문가 과정

강사-INNOVA LEE(이상훈)
Gccompil3r@gmail.com
학생-윤지완
Yoonjw789 @naver.com

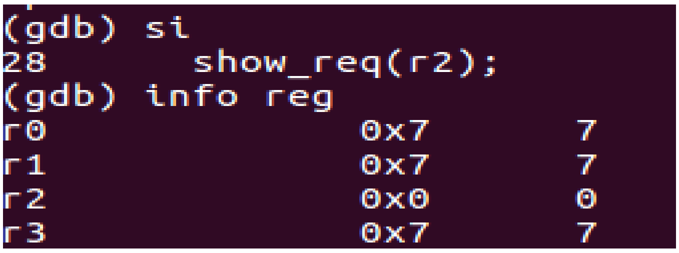
```
#include<stdio.h>

void show_req(unsigned int req)
{
int i;
for(i=3;i>=0;)
printf("%d",(req>>i--)&1);
printf("\n");
}

int main(void)
{
register unsigned  int r0 asm("r0");
register unsigned  int r1 asm("r1");
register unsigned  int r2 asm("r2");
register unsigned  int r3 asm("r3");
register unsigned  int r4 asm("r4");
register unsigned  int r5 asm("r5");



r0=7;
r1=7;
if(r0==r1)
{
r2=42;      0x101010
asm volatile("biceq r2,r3, #7");
}
show_req(r2);
return 0;
}
```



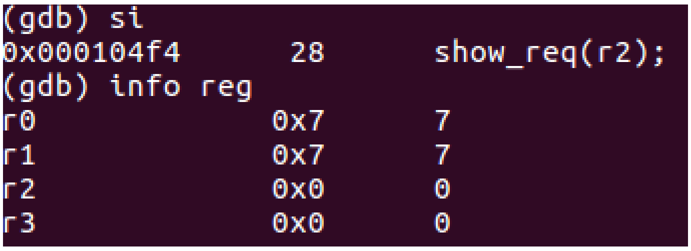
```
26        asm volatile("biceq r2,r3, #7");
(gdb) info reg
r0             0x7      7
r1             0x7      7
r2             0x2a     42
r3             0x7      7
r4             0x10508  66824
r5             0x0      0
r6             0x10340  66368
r7             0x0      0
r8             0x0      0
r9             0x0      0
r10            0xf67fe000       -159391744
r11            0xf6ffef14       -150999276
r12            0xf6ffef90       -150999152
sp             0xf6ffef10       0xf6ffef10
lr             0xf6686d14       -160928492
pc             0x104ec  0x104ec <main+32>
cpsr           0x60000010       1610612752
(gdb) 
```



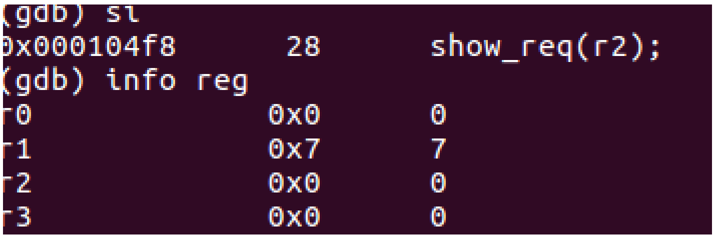
```
tic -L /usr/arm-linux-gnueab
```



```
(gdb) si
28        show_req(r2);
(gdb) info reg
r0                 0x7       7
r1                 0x7       7
r2                 0x0       0
r3                 0x7       7
```

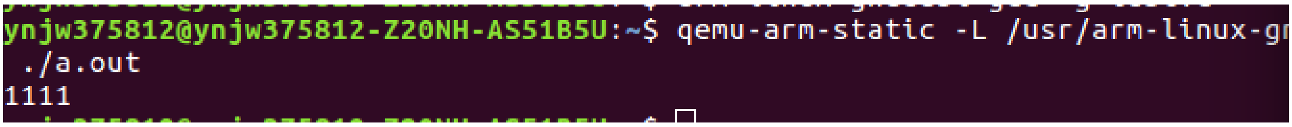


```
(gdb) si
0x000104f4      28      show_req(r2);
(gdb) info reg
r0              0x7       7
r1              0x7       7
r2              0x0       0
r3              0x0       0
```



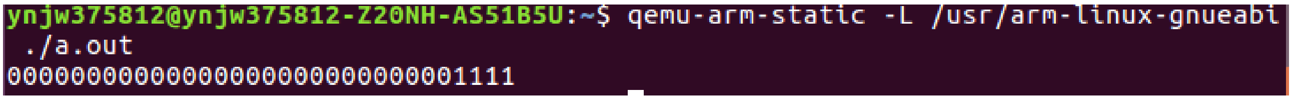
```
(gdb) si
0x000104f8      28      show_req(r2);
(gdb) info reg
r0              0x0       0
r1              0x7       7
r2              0x0       0
r3              0x0       0
```

```
#include<stdio.h>

void show_req(unsigned int req)
{
int i;
for(i=31;i>=0;)
printf("%d",(req>>i--)&1);
printf("\n");
}

int main(void)
{
register unsigned  int r0 asm("r0");
register unsigned  int r1 asm("r1");
register unsigned  int r2 asm("r2");
register unsigned  int r3 asm("r3");
register unsigned  int r4 asm("r4");
register unsigned  int r5 asm("r5");

r0=1;
r1=1;
r5=3;
if(r0==r1)
{
  r3=44;0x101100
// r5=3   0x000011
//           0x101111=47
asm volatile("orr r2,r3, r5");
}
show_req(r2);
return 0;
}
```



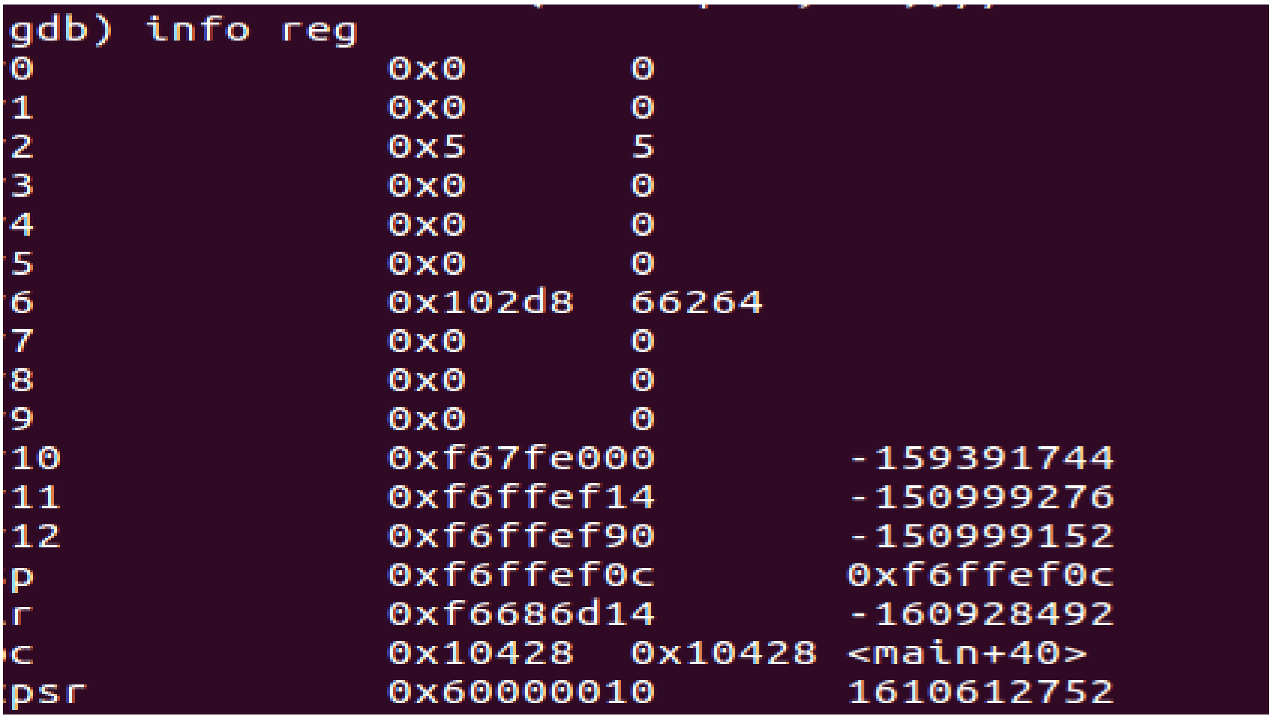
```
ynjw375812@ynjw375812-Z20NH-AS51B5U:~$ qemu-arm-static -L /usr/arm-linux-g
 ./a.out
1111
```

```
#include<stdio.h>

void show_req(unsigned int req)
{
int i;
for(i=31;i>=0;)
printf("%d",(req>>i--)&1);
printf("\n");
}

int main(void)
{
register unsigned  int r0 asm("r0");
register unsigned  int r1 asm("r1");
register unsigned  int r2 asm("r2");
register unsigned  int r3 asm("r3");
register unsigned  int r4 asm("r4");
register unsigned  int r5 asm("r5");

r1=10;0x00000000000000000000000000001010
r0=10; 0x00000000000000000000000000001010
if(r0==r1)
{
//r0=10; 0x00000000000000000000000000001010
r3=5;     0x00000000000000000000000000000101
             0x00000000000000000000000000001111
asm volatile("eors r1,r3, r0");
}
show_req(r1);
return 0;
}
```



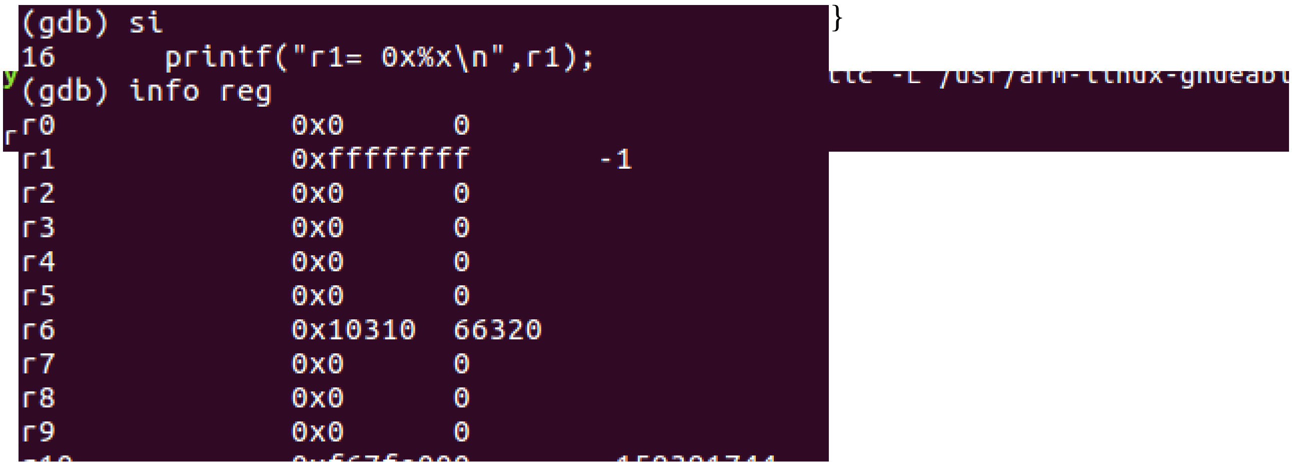
```
ynjw375812@ynjw375812-Z20NH-AS51B5U:~$ qemu-arm-static -L /usr/arm-linux-gnueabi
 ./a.out
00000000000000000000000000001111
```

```
#include<stdio.h>
void show_req(unsigned int req)
{
int i;
for(i=31;i>=0;)
{
printf("%d",(req>>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("cmp r0,r1");
asm volatile("mov r2, #5");//r2 에다가 5 를 넣는다.
asm volatile("tsteq r0,r2");//tst=AND 연산



return 0;
```



```
gdb) info reg
0              0x0        0
1              0x0        0
2              0x5        5
3              0x0        0
4              0x0        0
5              0x0        0
6              0x102d8    66264
7              0x0        0
8              0x0        0
9              0x0        0
10             0xf67fe000         -159391744
11             0xf6ffef14         -150999276
12             0xf6ffef90         -150999152
p              0xf6ffef0c         0xf6ffef0c
r              0xf6686d14         -160928492
c              0x10428    0x10428 <main+40>
psr            0x60000010         1610612752
```

```
#include<stdio.h>
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("cmp r0,r1");//cmp=eq zero flag 가 0 이여야 동작이 된다.

asm volatile("mvneq r1, #0");//xor 연산

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


return 0;
}
```



```
(gdb) si
16          printf("r1= 0x%x\n",r1);
(gdb) info reg                            llc -L /usr/arm-linux-gnueabi
r0             0x0        0
r1             0xffffffff         -1
r2             0x0        0
r3             0x0        0
r4             0x0        0
r5             0x0        0
r6             0x10310    66320
r7             0x0        0
r8             0x0        0
r9             0x0        0
r10            0xf67fe000         -159391744
```