ARM Assembly Exercise (1B)

Young Won Lim
7/16/16

Please send corrections (or suggestions) to youngwlim@hotmail.com.

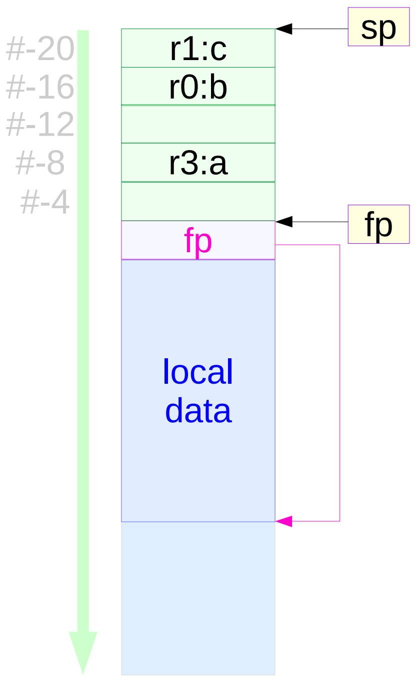
This document was produced by using LibreOffice.

Intermixing Source

```
arm-linux-gnueabi-gcc -g -c t1.c

arm-linux-gnueabi-objdump -S t1.o
```

[1] Add & Sutract

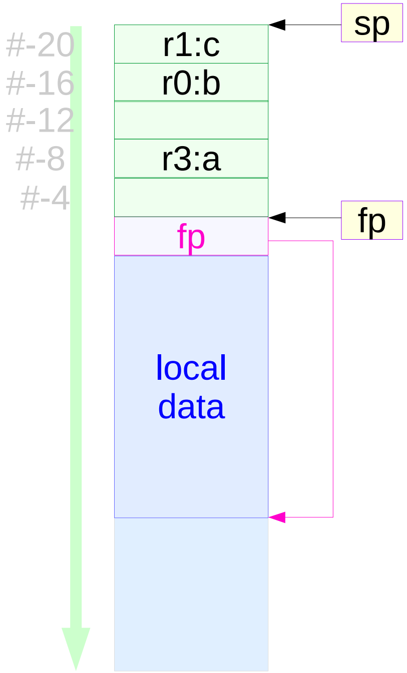
```
void t1(int b, int c) {

  int a;

  a = b + c;

}
```



```
00000000 <t1>:
void t1(int b, int c) {
   0:   e52db004     push{fp}     ; (str fp, [sp, #-4]!)
   4:   e28db000     add fp, sp, #0
   8:   e24dd014     sub sp, sp, #20
   c:   e50b0010     str r0, [fp, #-16]
  10:   e50b1014     str r1, [fp, #-20]

  int a;

  a = b + c;
  14:   e51b2010     ldr r2, [fp, #-16]
  18:   e51b3014     ldr r3, [fp, #-20]
  1c:   e0823003     add r3, r2, r3
  20:   e50b3008     str r3, [fp, #-8]

}
  24:   e28bd000     add sp, fp, #0
  28:   e8bd0800     ldmfd   sp!, {fp}
  2c:   e12fff1e     bx  lr
```

[2] Add & Sutract

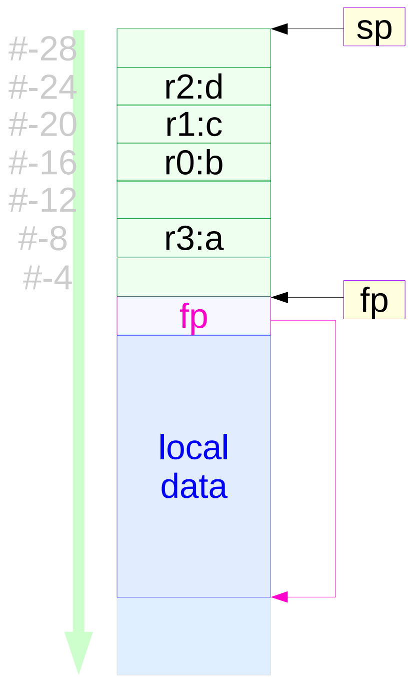
```
void t2(int b, int c) {

  int a;

  a = b - c;

}
```



```
00000000 <t1>:
void t1(int b, int c) {
   0:    e52db004     push{fp}     ; (str fp, [sp, #-4]!)
   4:    e28db000     add fp, sp, #0
   8:    e24dd014     sub sp, sp, #20
   c:    e50b0010     str r0, [fp, #-16]
  10:    e50b1014     str r1, [fp, #-20]

  int a;

  a = b - c;
  14:    e51b2010     ldr r2, [fp, #-16]
  18:    e51b3014     ldr r3, [fp, #-20]
  1c:    e0633002     rsb r3, r3, r2
  20:    e50b3008     str r3, [fp, #-8]

}
  24:    e28bd000     add sp, fp, #0
  28:    e8bd0800     ldmfd    sp!, {fp}
  2c:    e12fff1e     bx  lr
```

[3] Add & Sutract

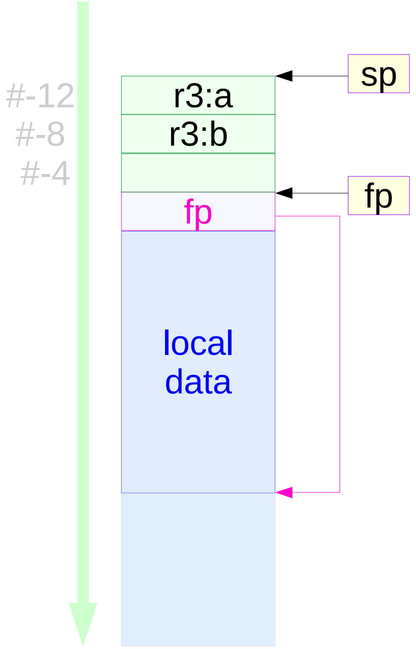
```
void t3(int b, int c, int d)
{

  int a;

  a = b + c - d;

}
```



```
00000000 <t3>:
void t3(int b, int c, int d) {
   0:   e52db004     push{fp}     ; (str fp, [sp, #-4]!)
   4:   e28db000     add fp, sp, #0
   8:   e24dd01c     sub sp, sp, #28
   c:   e50b0010     str r0, [fp, #-16]
  10:   e50b1014     str r1, [fp, #-20]
  14:   e50b2018     str r2, [fp, #-24]

  int a;

  a = b + c - d;
  18:   e51b2010     ldr r2, [fp, #-16]
  1c:   e51b3014     ldr r3, [fp, #-20]
  20:   e0822003     add r2, r2, r3
  24:   e51b3018     ldr r3, [fp, #-24]
  28:   e0633002     rsb r3, r3, r2
  2c:   e50b3008     str r3, [fp, #-8]

}
  30:   e28bd000     add sp, fp, #0
  34:   e8bd0800     ldmfd    sp!, {fp}
  38:   e12fff1e     bx   lr
```

[4] Constant

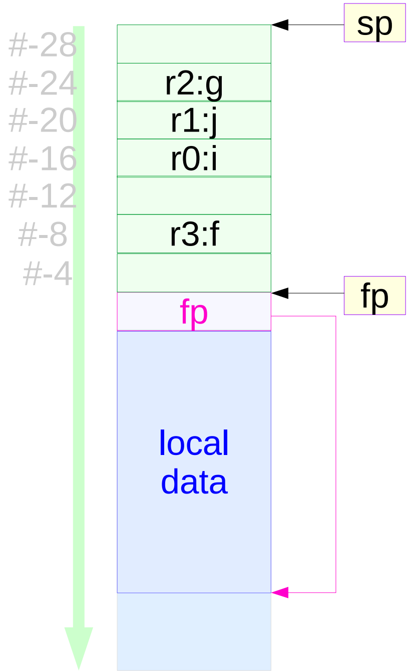
```
void t4(void) {

   int a = 0x4F3C;;
   int b = 0x6D5E4F3C;

}
```



```
00000000 <t4>:
void t4(void) {
   0:    e52db004     push{fp}     ; (str fp, [sp, #-4]!)
   4:    e28db000     add fp, sp, #0
   8:    e24dd00c     sub sp, sp, #12

  int a = 0x4F3C;;
   c:    e59f3014     ldr r3, [pc, #20]    ; 28 <t4+0x28>
  10:    e50b300c     str r3, [fp, #-12]
  int b = 0x6D5E4F3C;
  14:    e59f3010     ldr r3, [pc, #16]    ; 2c <t4+0x2c>
  18:    e50b3008     str r3, [fp, #-8]

}
  1c:    e28bd000     add sp, fp, #0
  20:    e8bd0800     ldmfd    sp!, {fp}
  24:    e12fff1e     bx  lr
  28:    00004f3c     .word    0x00004f3c
  2c:    6d5e4f3c     .word    0x6d5e4f3c
```

[5] if Statements

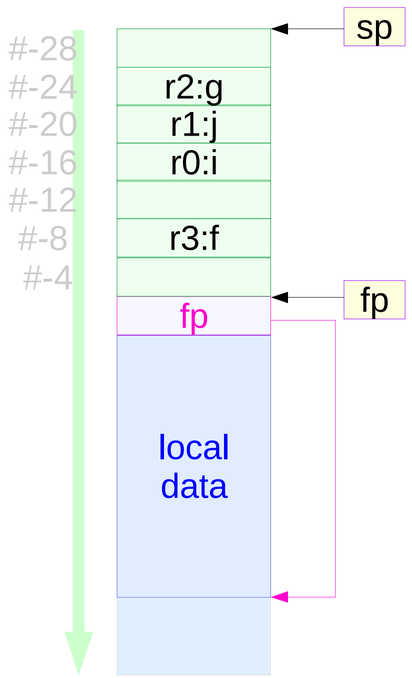
```
void t5(int i, int j, int g)
{
  int f;

  if (i == j) f = g + 1;
  f = f - 1;

}
```



```
00000000 <t5>:
void t5(int i, int j, int g) {
   0:   e52db004     push{fp}     ; (str fp, [sp, #-4]!)
   4:   e28db000     add fp, sp, #0
   8:   e24dd01c     sub sp, sp, #28
   c:   e50b0010     str r0, [fp, #-16]
  10:   e50b1014     str r1, [fp, #-20]
  14:   e50b2018     str r2, [fp, #-24]
  int f;

  if (i == j) f = g + 1;
  18:   e51b2010     ldr r2, [fp, #-16]
  1c:   e51b3014     ldr r3, [fp, #-20]
  20:   e1520003     cmp r2, r3
  24:   1a000002     bne 34 <t5+0x34>
  28:   e51b3018     ldr r3, [fp, #-24]
  2c:   e2833001     add r3, r3, #1
  30:   e50b3008     str r3, [fp, #-8]
  f = f - 1;
  34:   e51b3008     ldr r3, [fp, #-8]
  38:   e2433001     sub r3, r3, #1
  3c:   e50b3008     str r3, [fp, #-8]

}
  40:   e28bd000     add sp, fp, #0
  44:   e8bd0800     ldmfd   sp!, {fp}
  48:   e12fff1e     bx  lr
```

[6] if – else Statements

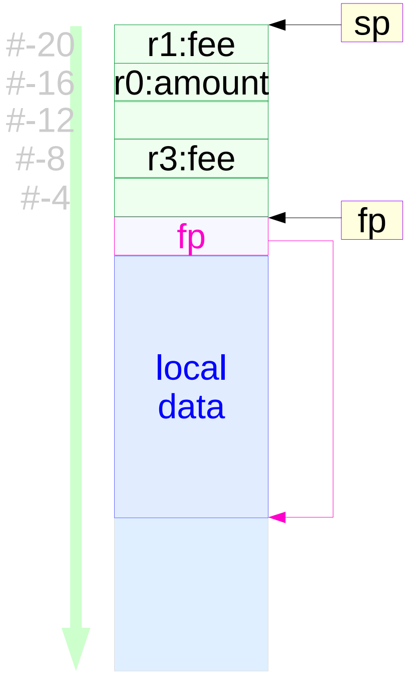
```
void t6(int i, int j, int g)
{
  int f;

  if (i == j) f = g + 1;
  else f = f - 1;

}
```



```
00000000 <t6>:
void t6(int i, int j, int g) {
   0:   e52db004     push{fp}     ; (str fp, [sp, #-4]!)
   4:   e28db000     add fp, sp, #0
   8:   e24dd01c     sub sp, sp, #28
   c:   e50b0010     str r0, [fp, #-16]
  10:   e50b1014     str r1, [fp, #-20]
  14:   e50b2018     str r2, [fp, #-24]
  int f;

  if (i == j) f = g + 1;
  18:   e51b2010     ldr r2, [fp, #-16]
  1c:   e51b3014     ldr r3, [fp, #-20]
  20:   e1520003     cmp r2, r3
  24:   1a000003     bne 38 <t6+0x38>
  28:   e51b3018     ldr r3, [fp, #-24]
  2c:   e2833001     add r3, r3, #1
  30:   e50b3008     str r3, [fp, #-8]
  34:   ea000002     b   44 <t6+0x44>
  else f = f - 1;
  38:   e51b3008     ldr r3, [fp, #-8]
  3c:   e2433001     sub r3, r3, #1
  40:   e50b3008     str r3, [fp, #-8]

}
  44:   e28bd000     add sp, fp, #0
  48:   e8bd0800     ldmfd   sp!, {fp}
  4c:   e12fff1e     bx  lr
```

[7] switch Statements (1)

```
void t7(int amount, int fee) {

   switch (amount) {
     case 20  : fee = 2; break;
     case 50  : fee = 3; break;
     default  : fee = 0;
  }
}
```



```
00000000 <t7>:
void t7(int amount, int fee) {
   0:    e52db004     push{fp}     ; (str
fp, [sp, #-4]!)
   4:    e28db000     add  fp, sp, #0
   8:    e24dd014     sub  sp, sp, #20
   c:    e50b0010     str  r0, [fp, #-16]
  10:    e50b1014     str  r1, [fp, #-20]

 }
}
  4c:    e28bd000     add  sp, fp, #0
  50:    e8bd0800     ldmfd    sp!, {fp}
  54:    e12fff1e     bx   lr
```

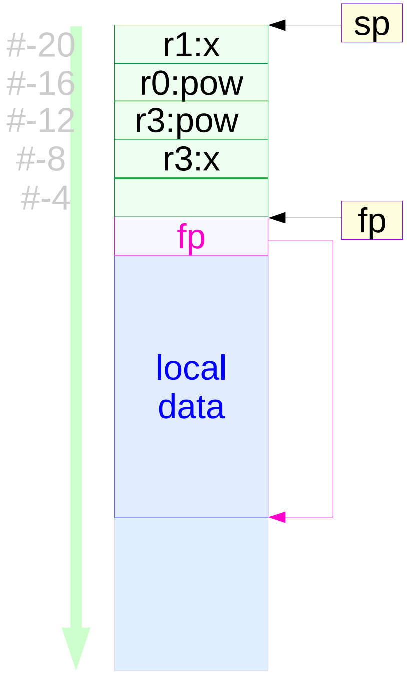
[7] switch Statements (2)

```
void t7(int amount, int fee) {

  switch (amount) {
    case 20  : fee = 2; break;
    case 50  : fee = 3; break;
    default  : fee = 0;
 }
}
```

```
   switch (amount) {
   14:    e51b3010     ldr r3, [fp, #-16]
   18:    e3530014     cmp r3, #20
   1c:    0a000002     beq 2c <t7+0x2c>
   20:    e3530032     cmp r3, #50  ; 0x32
   24:    0a000003     beq 38 <t7+0x38>
   28:    ea000005     b   44 <t7+0x44>
     case 20  : fee = 2; break;
   2c:    e3a03002     mov r3, #2
   30:    e50b3008     str r3, [fp, #-8]
   34:    ea000004     b   4c <t7+0x4c>
     case 50  : fee = 3; break;
   38:    e3a03003     mov r3, #3
   3c:    e50b3008     str r3, [fp, #-8]
   40:    ea000001     b   4c <t7+0x4c>
     default  : fee = 0;
   44:    e3a03000     mov r3, #0
   48:    e50b3008     str r3, [fp, #-8]
  }
 }
   4c:    e28bd000     add sp, fp, #0
   50:    e8bd0800     ldmfd   sp!, {fp}
   54:    e12fff1e     bx  lr
```

[8] while Loop (1)

```
void t8(int pow, int x) {

   pow = 1;
   x = 1;

   while (pow != 128) {
     pow = pow * 2;
     x   = x   + 1;
   }
}
```



```
00000000 <t8>:
void t8(int pow, int x) {
   0:    e52db004     push{fp}     ; (str
fp, [sp, #-4]!)
   4:    e28db000     add fp, sp, #0
   8:    e24dd014     sub sp, sp, #20
   c:    e50b0010     str r0, [fp, #-16]
  10:    e50b1014     str r1, [fp, #-20]

  pow = 1;
  14:    e3a03001     mov r3, #1
  18:    e50b300c     str r3, [fp, #-12]
  x = 1;
  1c:    e3a03001     mov r3, #1
  20:    e50b3008     str r3, [fp, #-8]




  }
}
  4c:    e28bd000     add sp, fp, #0
  50:    e8bd0800     ldmfd    sp!, {fp}
  54:    e12fff1e     bx  lr
```

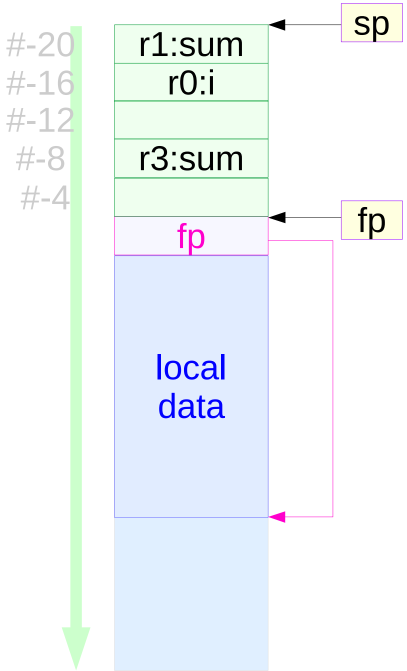
[8] while Loop (2)

```
void t8(void) {

 int pow = 1;
 int   x = 1;

 while (pow != 128) {
   pow = pow * 2;
   x   = x   + 1;
 }
}
```

```
 while (pow != 128) {
 24:    ea000005     b    40 <t8+0x40>
   pow = pow * 2;
 28:    e51b300c     ldr r3, [fp, #-12]
 2c:    e1a03083     lsl r3, r3, #1
 30:    e50b300c     str r3, [fp, #-12]
   x   = x   + 1;
 34:    e51b3008     ldr r3, [fp, #-8]
 38:    e2833001     add r3, r3, #1
 3c:    e50b3008     str r3, [fp, #-8]
void t8(int pow, int x) {

 pow = 1;
 x = 1;

 while (pow != 128) {
 40:    e51b300c     ldr r3, [fp, #-12]
 44:    e3530080     cmp r3, #128 ; 0x80
 48:    1afffff6     bne 28 <t8+0x28>
   pow = pow * 2;
   x   = x   + 1;
 }
}
```

[9] for Loop (1)

```
void t9(int i, int sum) {
  sum = 0;

  for (i=0; i!=10; i=i+1) {
    sum = sum + i;
  }
}
```



```
00000000 <t9>:
void t9(int i, int sum) {
   0:    e52db004     push{fp}     ; (str
fp, [sp, #-4]!)
   4:    e28db000     add fp, sp, #0
   8:    e24dd014     sub sp, sp, #20
   c:    e50b0010     str r0, [fp, #-16]
  10:    e50b1014     str r1, [fp, #-20]
  sum = 0;
  14:    e3a03000     mov r3, #0
  18:    e50b3008     str r3, [fp, #-8]





  50:    e28bd000     add sp, fp, #0
  54:    e8bd0800     ldmfd    sp!, {fp}
  58:    e12fff1e     bx  lr
```

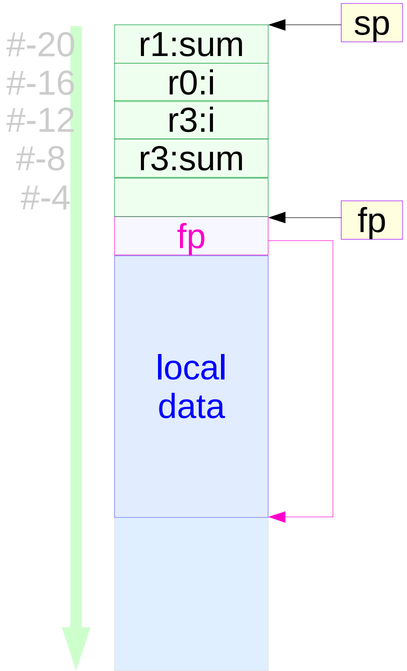
[9] for Loop (2)

```
void t9(int i, int sum) {
  sum = 0;

  for (i=0; i!=10; i=i+1) {
    sum = sum + i;
  }
}
```

```
  for (i=0; i!=10; i=i+1) {
  1c:    e3a03000     mov r3, #0
  20:    e50b300c     str r3, [fp, #-12]
  24:    ea000006     b   44 <t9+0x44>
    sum = sum + i;
  28:    e51b2008     ldr r2, [fp, #-8]
  2c:    e51b300c     ldr r3, [fp, #-12]
  30:    e0823003     add r3, r2, r3
  34:    e50b3008     str r3, [fp, #-8]
void t9(int i, int sum) {
  sum = 0;

  for (i=0; i!=10; i=i+1) {
  38:    e51b300c     ldr r3, [fp, #-12]
  3c:    e2833001     add r3, r3, #1
  40:    e50b300c     str r3, [fp, #-12]
  44:    e51b300c     ldr r3, [fp, #-12]
  48:    e353000a     cmp r3, #10
  4c:    1affff5      bne 28 <t9+0x28>
    sum = sum + i;
  }
}
```

[10] for – while Loop (1)

```
void t10(int i, int sum) {
  sum = 0;
  i   = 0;

  while (i!=10) {
    sum = sum + i;
    i = i + 1;
  }
}
```



```
00000000 <t10>:
void t10(int i, int sum) {
   0:    e52db004     push{fp}     ; (str
fp, [sp, #-4]!)
   4:    e28db000     add  fp, sp, #0
   8:    e24dd014     sub  sp, sp, #20
   c:    e50b0010     str  r0, [fp, #-16]
  10:    e50b1014     str  r1, [fp, #-20]
  sum = 0;
  14:    e3a03000     mov  r3, #0
  18:    e50b3008     str  r3, [fp, #-8]
  i   = 0;
  1c:    e3a03000     mov  r3, #0
  20:    e50b300c     str  r3, [fp, #-12]




  50:    e28bd000     add  sp, fp, #0
  54:    e8bd0800     ldmfd    sp!, {fp}
  58:    e12fff1e     bx   lr
```

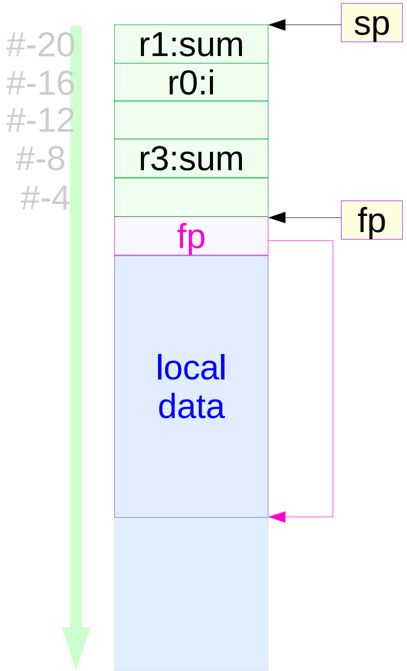
[10] for – while Loop (2)

```
void t10(int i, int sum) {
  sum = 0;
  i   = 0;

  while (i!=10) {
    sum = sum + i;
    i = i + 1;
  }
}
```

```
  while (i!=10) {
  24:   ea000006    b   44 <t10+0x44>
    sum = sum + i;
  28:   e51b2008    ldr r2, [fp, #-8]
  2c:   e51b300c    ldr r3, [fp, #-12]
  30:   e0823003    add r3, r2, r3
  34:   e50b3008    str r3, [fp, #-8]
    i = i + 1;
  38:   e51b300c    ldr r3, [fp, #-12]
  3c:   e2833001    add r3, r3, #1
  40:   e50b300c    str r3, [fp, #-12]
void t10(int i, int sum) {
  sum = 0;
  i   = 0;

  while (i!=10) {
  44:   e51b300c    ldr r3, [fp, #-12]
  48:   e353000a    cmp r3, #10
  4c:   1affffff5   bne 28 <t10+0x28>
    sum = sum + i;
    i = i + 1;
  }
}
```

[11] for – from 1 Loop (1)

```
void t11(int i, int sum) {
  sum = 0;
  i   = 0;

  for (i=1; i<101; i=i*2) {
    sum = sum + i;
    i = i + 1;
  }
}
```



```
00000000 <t11>:
void t11(int i, int sum) {
   0:    e52db004     push{fp}     ; (str
fp, [sp, #-4]!)
   4:    e28db000     add fp, sp, #0
   8:    e24dd014     sub sp, sp, #20
   c:    e50b0010     str r0, [fp, #-16]
  10:    e50b1014     str r1, [fp, #-20]
  sum = 0;
  14:    e3a03000     mov r3, #0
  18:    e50b3008     str r3, [fp, #-8]
  i   = 0;
  1c:    e3a03000     mov r3, #0
  20:    e50b300c     str r3, [fp, #-12]



  58:    e28bd000     add sp, fp, #0
  5c:    e8bd0800     ldmfd    sp!, {fp}
  60:    e12fff1e     bx  lr
```

[11] for – from 1 Loop (2)

```
void t11(int i, int sum) {
  sum = 0;
  i   = 0;

  for (i=1; i<101; i=i*2) {
    sum = sum + i;
    i = i + 1;
  }
}
```

```
  for (i=1; i<101; i=i*2) {
  24:    e3a03001     mov r3, #1
  28:    e50b300c     str r3, [fp, #-12]
  2c:    ea000006     b   4c <t11+0x4c>
    sum = sum + i;
  30:    e51b2008     ldr r2, [fp, #-8]
  34:    e51b300c     ldr r3, [fp, #-12]
  38:    e0823003     add r3, r2, r3
  3c:    e50b3008     str r3, [fp, #-8]
void t11(int i, int sum) {
  sum = 0;
  i   = 0;

  for (i=1; i<101; i=i*2) {
  40:    e51b300c     ldr r3, [fp, #-12]
  44:    e1a03083     lsl r3, r3, #1
  48:    e50b300c     str r3, [fp, #-12]
  4c:    e51b300c     ldr r3, [fp, #-12]
  50:    e3530064     cmp r3, #100 ; 0x64
  54:    daffff f5     ble 30 <t11+0x30>
    sum = sum + i;
  }
}
```

[12] Arrays

```
void t12(void) {
  int array[5];

  array[0] = array[0] * 8;
  array[1] = array[1] + 8;
}
```

```
00000000 <t12>:
void t12(void) {
   0:    e52db004     push{fp}     ; (str
fp, [sp, #-4]!)
   4:    e28db000     add fp, sp, #0
   8:    e24dd01c     sub sp, sp, #28
  int array[5];

  array[0] = array[0] * 8;
   c:    e51b3018     ldr r3, [fp, #-24]
  10:    e1a03183     lsl r3, r3, #3
  14:    e50b3018     str r3, [fp, #-24]
  array[1] = array[1] + 8;
  18:    e51b3014     ldr r3, [fp, #-20]
  1c:    e2833008     add r3, r3, #8
  20:    e50b3014     str r3, [fp, #-20]
}
  24:    e28bd000     add sp, fp, #0
  28:    e8bd0800     ldmfd    sp!, {fp}
  2c:    e12fff1e     bx  lr
```

[13] Large Arrays (1)

```
void t13(int i) {
  int array[1000];

  for (i=0; i<1000; i=i+1) {
    array[i] = array[i] * 8;
  }
}
```

```
00000000 <t13>:
void t13(int i) {
   0:    e52db004     push{fp}     ; (str fp, [sp, #-4]!)
   4:    e28db000     add fp, sp, #0
   8:    e24ddefb     sub sp, sp, #4016    ; 0xfb0
   c:    e24dd004     sub sp, sp, #4
  10:    e50b0fb0     str r0, [fp, #-4016] ; 0xfb0
  int array[1000];

  for (i=0; i<1000; i=i+1) {
  14:    e3a03000     mov r3, #0
  18:    e50b3fa8     str r3, [fp, #-4008] ; 0xfa8
  1c:    ea000011     b   68 <t13+0x68>
```

[13] Large Arrays (2)

```
void t13(int i) {
  int array[1000];

  for (i=0; i<1000; i=i+1) {
    array[i] = array[i] * 8;
  }
}
```

```
    array[i] = array[i] * 8;
20:    e51b2fa8     ldr r2, [fp, #-4008] ; 0xfa8
24:    e59f3058     ldr r3, [pc, #88]    ; 84 <t13+0x84>
28:    e1a02102     lsl r2, r2, #2
2c:    e24b0004     sub r0, fp, #4
30:    e0802002     add r2, r0, r2
34:    e0823003     add r3, r2, r3
38:    e5933000     ldr r3, [r3]
3c:    e1a02183     lsl r2, r3, #3
40:    e51b1fa8     ldr r1, [fp, #-4008] ; 0xfa8
44:    e59f3038     ldr r3, [pc, #56]    ; 84 <t13+0x84>
48:    e1a01101     lsl r1, r1, #2
4c:    e24b0004     sub r0, fp, #4
50:    e0801001     add r1, r0, r1
54:    e0813003     add r3, r1, r3
58:    e5832000     str r2, [r3]
```

[13] Large Arrays (3)

```
void t13(int i) {
  int array[1000];

  for (i=0; i<1000; i=i+1) {
    array[i] = array[i] * 8;
  }
}
```

```
void t13(int i) {
  int array[1000];

  for (i=0; i<1000; i=i+1) {
  5c:   e51b3fa8    ldr r3, [fp, #-4008] ; 0xfa8
  60:   e2833001    add r3, r3, #1
  64:   e50b3fa8    str r3, [fp, #-4008] ; 0xfa8
  68:   e51b2fa8    ldr r2, [fp, #-4008] ; 0xfa8
  6c:   e59f3014    ldr r3, [pc, #20]    ; 88 <t13+0x88>
  70:   e1520003    cmp r2, r3
  74:   daffffe9    ble 20 <t13+0x20>
    array[i] = array[i] * 8;
  }
}
  78:   e28bd000    add sp, fp, #0
  7c:   e8bd0800    ldmfd   sp!, {fp}
  80:   e12fff1e    bx  lr
  84:   fffff060    .word   0xfffff060
  88:   000003e7    .word   0x000003e7
```

[14] Procedure Calls

```
int func(void) {
  return 0;
}

int main(void) {
  func();
  return 0;
}
```

```
00000000 <func>:
int func(void) {
   0:   e52db004     push{fp}     ; (str fp, [sp, #-4]!)
   4:   e28db000     add fp, sp, #0
  return 0;
   8:   e3a03000     mov r3, #0
}
   c:   e1a00003     mov r0, r3
  10:   e28bd000     add sp, fp, #0
  14:   e8bd0800     ldmfd    sp!, {fp}
  18:   e12fff1e     bx  lr

0000001c <main>:

int main(void) {
  1c:   e92d4800     push{fp, lr}
  20:   e28db004     add fp, sp, #4
  func();
  24:   ebfffffe     bl  0 <func>
  return 0;
  28:   e3a03000     mov r3, #0
}
  2c:   e1a00003     mov r0, r3
  30:   e8bd8800     pop {fp, pc}
```

[15] Argument Passing - (A)

```
int foo(int f, int g,
        int h, int i) {
  int result;

  result = (f+g)-(h+i);
  return result;
}

int t15(void) {
  int y;

  y = foo(2,3,4,5);
  return 0;
}
```

```
00000000 <foo>:
int foo(int f, int g, int h, int i) {
   0:   e52db004     push{fp}    ; (str fp, [sp, #-4]!)
   4:   e28db000     add fp, sp, #0
   8:   e24dd01c     sub sp, sp, #28
   c:   e50b0010     str r0, [fp, #-16]
  10:   e50b1014     str r1, [fp, #-20]
  14:   e50b2018     str r2, [fp, #-24]
  18:   e50b301c     str r3, [fp, #-28]
  int result;

  result = (f+g)-(h+i);
  1c:   e51b2010     ldr r2, [fp, #-16]
  20:   e51b3014     ldr r3, [fp, #-20]
  24:   e0822003     add r2, r2, r3
  28:   e51b1018     ldr r1, [fp, #-24]
  2c:   e51b301c     ldr r3, [fp, #-28]
  30:   e0813003     add r3, r1, r3
  34:   e0633002     rsb r3, r3, r2
  38:   e50b3008     str r3, [fp, #-8]
  return result;
  3c:   e51b3008     ldr r3, [fp, #-8]
}
  40:   e1a00003     mov r0, r3
  44:   e28bd000     add sp, fp, #0
  48:   e8bd0800     ldmfd    sp!, {fp}
  4c:   e12fff1e     bx  lr
```

[15] Argument Passing - (B)

```
int foo(int f, int g,
        int h, int i) {
  int result;

  result = (f+g)-(h+i);
  return result;
}

int t15(void) {
  int y;

  y = foo(2,3,4,5);
  return 0;
}
```

```
int t15(void) {
  50:    e92d4800     push{fp, lr}
  54:    e28db004     add fp, sp, #4
  58:    e24dd008     sub sp, sp, #8
  int y;

  y = foo(2,3,4,5);
  5c:    e3a00002     mov r0, #2
  60:    e3a01003     mov r1, #3
  64:    e3a02004     mov r2, #4
  68:    e3a03005     mov r3, #5
  6c:    ebfffffe     bl  0 <foo>
  70:    e50b0008     str r0, [fp, #-8]
  return 0;
  74:    e3a03000     mov r3, #0
}
  78:    e1a00003     mov r0, r3
  7c:    e24bd004     sub sp, fp, #4
  80:    e8bd8800     pop {fp, pc}
```

ELF

References

[1] Essential C, Nick Parlante
[2] Efficient C Programming, Mark A. Weiss
[3] C A Reference Manual, Samuel P. Harbison & Guy L. Steele Jr.
[4] C Language Express, I. K. Chun
[5] “A Whirlwind Tutorial on Creating Really Teensy ELF Executables for Linux“
http://cseweb.ucsd.edu/~ricko/CSE131/teensyELF.htm
[6] http://en.wikipedia.org
[7] http://www.muppetlabs.com/~breadbox/software/tiny/teensy.html
[8] http://csapp.cs.cmu.edu/public/ch7-preview.pdf