STACKS on the ARM

stack-calledfunc.c:

```
int *
calledfunc1 (int *a, int *b, int *c, int *d, int *e, int *f, int *g, int *h)
{
        return a;
}

int *
calledfunc2 (int *a, int *b, int *c, int *d, int *e, int *f, int *g, int *h)
{
        return b;
}

int *
calledfunc3 (int *a, int *b, int *c, int *d, int *e, int *f, int *g, int *h)
{
        return e;
}

int *
calledfunc4 (int *a, int *b, int *c, int *d, int *e, int *f, int *g, int *h)
{
        return (int *)((int)a | (int)b | (int)c | (int)d | (int)e | (int)f | (int)g | (int)h);
}

int *
calledfunc5 (int *a, int *b, int *c, int *d, int *e, int *f, int *g, int *h)
{
        return dofunction(e, f, g, h);
}
```

stack-calledfunc.o:

```
Disassembly of section .text:

00000000 <calledfunc1>:
   0:    e12fff1e bx   lr

00000004 <calledfunc2>:
   4:    e1a00001 mov  r0, r1
   8:    e12fff1e bx   lr

0000000c <calledfunc3>:
   c:    e59d0000 ldr  r0, [sp]
  10:    e12fff1e bx   lr

00000014 <calledfunc4>:
  14:    e1811000 orr  r1, r1, r0
  18:    e1822001 orr  r2, r2, r1
  1c:    e1833002 orr  r3, r3, r2
  20:    e59d2000 ldr  r2, [sp]
  24:    e1821003 orr  r1, r2, r3
  28:    e59d3004 ldr  r3, [sp, #4]
  2c:    e1831001 orr  r1, r3, r1
  30:    e59d3008 ldr  r3, [sp, #8]
  34:    e59d000c ldr  r0, [sp, #12]
  38:    e1831001 orr  r1, r3, r1
  3c:    e1800001 orr  r0, r0, r1
  40:    e12fff1e bx   lr

00000044 <calledfunc5>:
  44:    e92d4008 push {r3, lr}
  48:    e28d0008 add  r0, sp, #8
  4c:    e890000f ldm  r0, {r0, r1, r2, r3}
  50:    ebfffffe bl   0 <dofunction>
  54:    e8bd4008 pop  {r3, lr}
  58:    e12fff1e bx   lr
```

stack-caller-1.c:

```
void
topfunc( void )
{
     int a, b, c, d, e, f, g, h, i, j;

     calledfunc(&i, &j);

     return a + j;
}
```

stack-caller-1.o:

```
00000000 <topfunc>:
   0:    e52de004 push {lr}     ; (str lr, [sp, #-4]!)
   4:    e24dd00c sub  sp, sp, #12
   8:    e1a0000d mov  r0, sp
   c:    e28d1004 add  r1, sp, #4
  10:    ebffffe bl   0 <calledfunc2>
  14:    e59d0004 ldr  r0, [sp, #4]
  18:    e28dd00c add  sp, sp, #12
  1c:    e49de004 pop  {lr}     ; (ldr lr, [sp], #4)
  20:    e12fff1e bx   lr
```

stack-caller-2.c:

```
void
topfunc( void )
{
     int a, b, c, d, e, f, g, h, i, j;

     calledfunc(&a, &b, &c, &d, &e, &f, &g, &h);

     calledfunc(&i, &j);

     return a + j;
}
```

stack-caller-2.o:

```
Disassembly of section .text:

00000000 <topfunc>:
   0:    e52de004 push {lr}     ; (str lr, [sp, #-4]!)
   4:    e24dd03c sub  sp, sp, #60   ; 0x3c
   8:    e28d0020 add  r0, sp, #32
   c:    e58d0000 str  r0, [sp]
  10:    e28d0024 add  r0, sp, #36   ; 0x24
  14:    e58d0004 str  r0, [sp, #4]
  18:    e28d0028 add  r0, sp, #40   ; 0x28
  1c:    e58d0008 str  r0, [sp, #8]
  20:    e28d002c add  r0, sp, #44   ; 0x2c
  24:    e28d301c add  r3, sp, #28
  28:    e28d1014 add  r1, sp, #20
  2c:    e28d2018 add  r2, sp, #24
  30:    e58d000c str  r0, [sp, #12]
  34:    e28d0010 add  r0, sp, #16
  38:    ebffffe bl   0 <calledfunc>
  3c:    e28d0030 add  r0, sp, #48   ; 0x30
  40:    e28d1034 add  r1, sp, #52   ; 0x34
  44:    ebffffe bl   0 <calledfunc>
  48:    e59d3034 ldr  r3, [sp, #52] ; 0x34
  4c:    e59d0010 ldr  r0, [sp, #16]
  50:    e0800003 add  r0, r0, r3
  54:    e28dd03c add  sp, sp, #60   ; 0x3c
  58:    e49de004 pop  {lr}     ; (ldr lr, [sp], #4)
  5c:    e12fff1e bx   lr
```