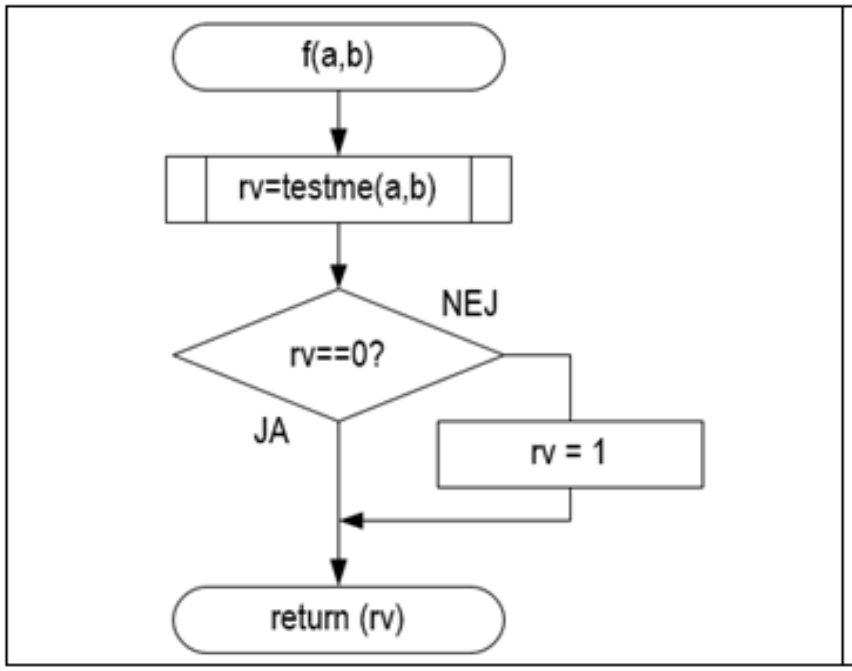
Registerspill

Problem: Det register man behöver är upptaget.
Lösning: Man lagrar temporärt registrets innehåll på annan plats (annat register eller i minnet), detta kallas ”registerspill”.

Exempel:
Funktionen `int testme( int a, int b )` är definierad.
Visa hur följande funktion kan kodas i assemblerspråk:

```
int f ( int x, int y )
{
if ( testme( x,y ) )
   return 1;
return 0;
}
```

Vi löser på tavlan...




```
f:
     PUSH {LR}
     BL   testme
     CMP  R0,#0
     BEQ  .L1
     MOV  R0,#1
.L1:
     POP  {PC}
```

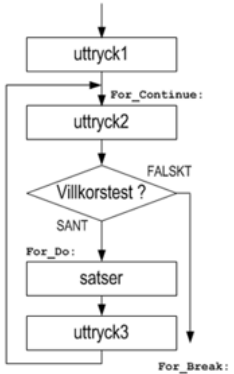
Registeranvändning med lokala variabler

Exempel:
Funktionen `int g( int )` är definierad.
Visa hur följande funktion kan kodas i assemblerspråk:

```
int f( int val )
{
    int i;
    int bits = 0;

    for( i=0 ; i< val; i++ )
    {
       bits = bits | g(i);
    }
    return bits;
}
```

Vi löser på tavlan...




```
@ Registerallokering:
@ R0, parameter och returvärde
@ R4 ”i”
@ R5 ”bits”
@ R6 ”val” spill från R0

f:
      PUSH        {R4,R5,R6,LR}
@ 'prolog'
      MOV         R6,R0             @ spillregister R6
      MOV         R5,#0             @ bits = 0;
@ 'uttryck 1'
      MOV         R4,#0             @ i = 0;
@ 'For_continue' - 'uttryck 2'
.L1:
      CMP         R4,R6             @ i - val
      BGE         .L2               @ i < val, komplementvillkor
@ 'satser'
      MOV         R0,R4
      BL          g                 @ g(i);
      ORR         R5,R5,R0          @ bits = bits | g(i);
@ 'uttryck 3'
      ADD         R4,R4,#1          @ i++;
      B           .L1
@ 'For_Break'
.L2:
      MOV         R0,R5             @ ”bits” -> R0 (returvärde)
      POP         {R4,R5,R6,PC}
```