

Program Execution

Networks and Embedded Software

Module 4.1.3 (optional)

by Wolfgang Neff

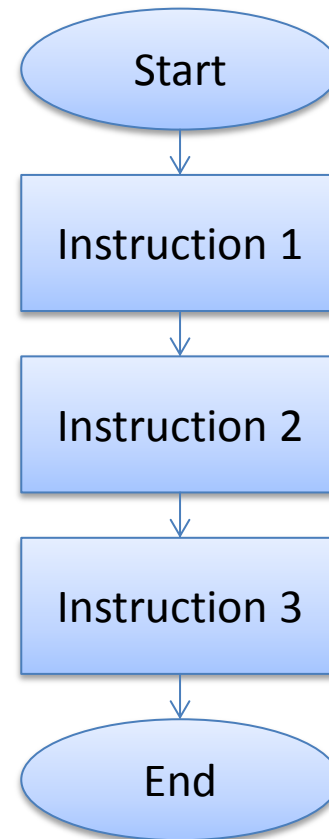
Control Flow (1)

- Sequence
 - One instruction at a time
 - One instruction after the other
 - From the beginning to the end

```
int a = 1;  
int b = 2;  
int c = a+b;
```

→

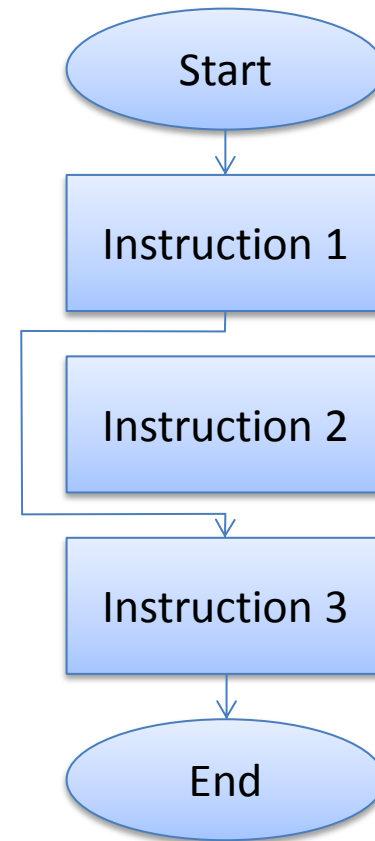
```
st a,=1  
st b,=2  
ld R0,a  
ld R1,b  
add R0,R1  
st c,R0
```



Control Flow (2)

- Jumps
 - Modifies control flow
 - Forward jumps skip code
 - Backward jumps make loops

a = 1;	→	st a,=1
goto label;		jmp label
b = 2;		st b,=2
label: c = 3;		label: st c,=3



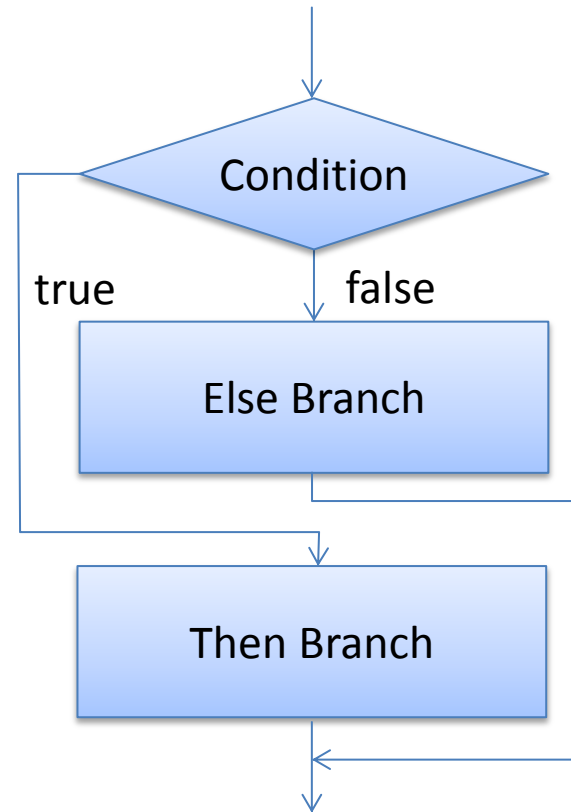
Control Flow (3)

- Branches
 - Conditional jumps
 - Make alternatives

```
if (n<5) {  
    n++;  
}  
else {  
    n=5;  
}
```

→

```
    cmp n,=5  
    br.lt then  
else: st a,=5  
      jmp end  
then: inc a  
end:  nop
```



Control Flow (4)

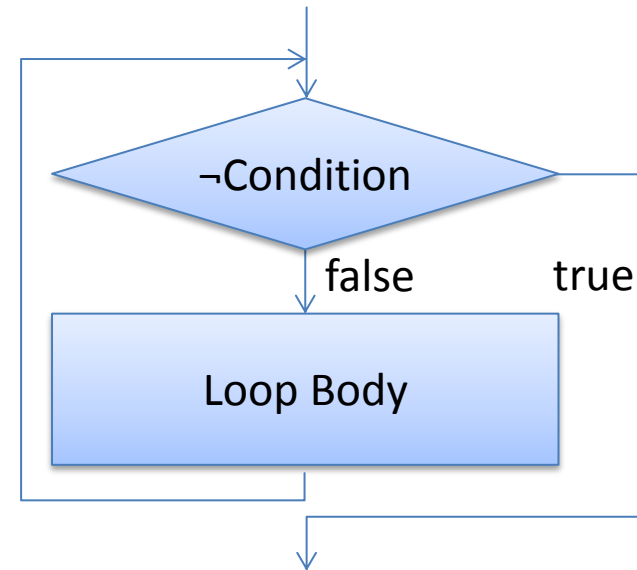
- Loops

- Backward branches.

- Types

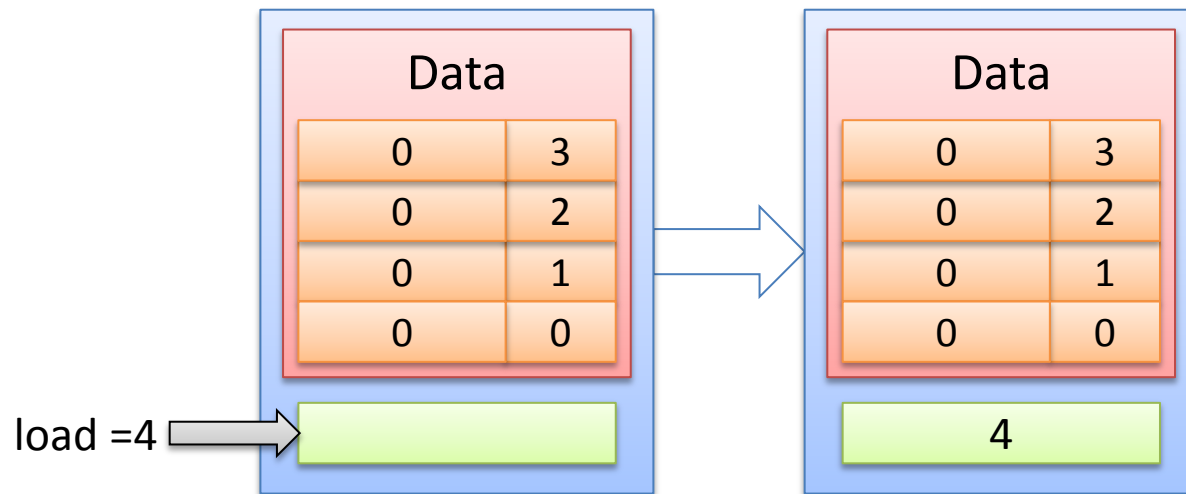
- While loop
- For loop

```
i=0;
while (i<10) {
  i++;
}
      ↗
loop: cmp R0,#10
      br.ge end
      inc R0
      jmp
end:  nop
```



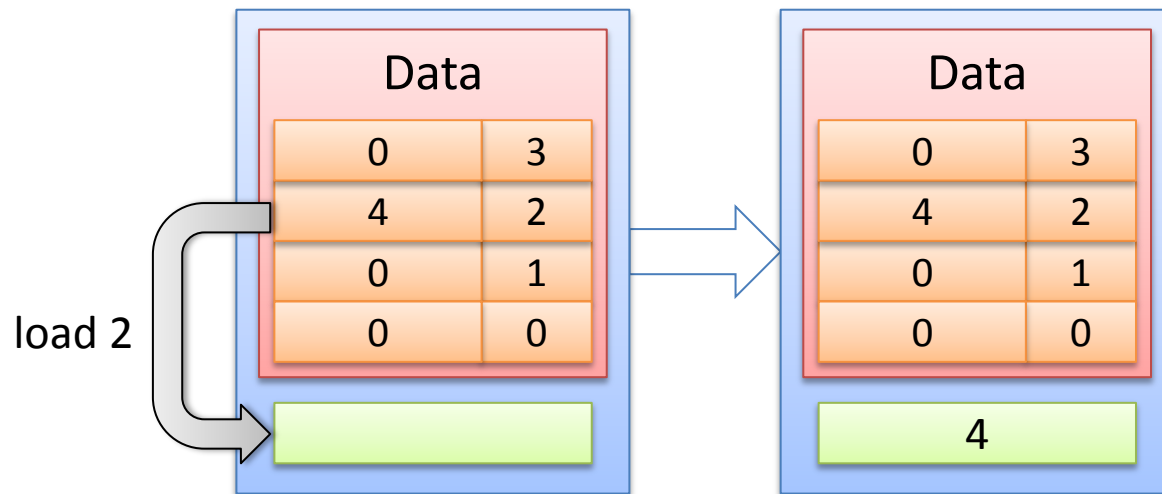
Addressing Modes (1)

- Immediate
 - Data immediately stored in register



Addressing Modes (2)

- Direct
 - Data loaded from a memory cell



Addressing Modes (3)

- Indirect
 - Data loaded from a cell another cell points to

