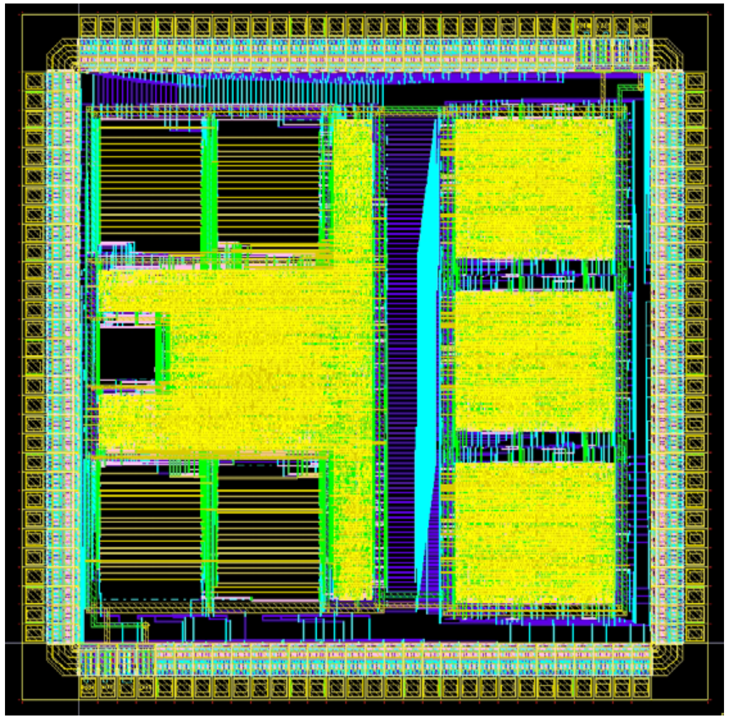
Digital Systems and Microprocessor Design (H7068)



9.2. Jumps and loops

Daniel Roggen
d.roggen@sussex.ac.uk

Content

- Loops with unconditional jumps
- Conditional jumps
- Conditional loops
- C-style loops to assembler

Loop with unconditional jump

- Unconditional jumps: changes the value of PC to destination
  - jmp   dst

- To do forever a task
  - Polling-based event loop
  - sensing-actuation loop

- Example:
  - Read sensors
  - Compute motor speed
  - Set motor speed

Loop with unconditional jump

```
PC    Adr   Instr                  RA    RB    RC    RD

->    00    mov   ra,3h            0     0     0     0
      02    sub   ra,1h
      04    jmp   2h
      06    ???
```

Loop with unconditional jump

PC	Adr	Instr	RA	RB	RC	RD
	00	mov ra,3h	3	0	0	0
->	02	sub ra,1h				
	04	jmp 2h				
	06	???				

Loop with unconditional jump

PC	Adr	Instr		RA	RB	RC	RD
	00	mov	ra,3h	2	0	0	0
	02	sub	ra,1h				
->	04	jmp	2h				
	06	???					

Loop with unconditional jump

```
PC    Adr   Instr               RA    RB    RC    RD

      00    mov   ra,3h         2     0     0     0
->    02    sub   ra,1h
      04    jmp   2h
      06    ???
```

Loop with unconditional jump

PC	Adr	Instr		RA	RB	RC	RD
	00	mov	ra,3h	1	0	0	0
	02	sub	ra,1h				
->	04	jmp	2h				
	06	???					

Loop with unconditional jump

```
PC     Adr    Instr                  RA     RB     RC     RD

       00     mov   ra,3h            1      0      0      0
->     02     sub   ra,1h
       04     jmp   2h
       06     ???
```

Loop with unconditional jump

PC	Adr	Instr		RA	RB	RC	RD
	00	mov	ra,3h	0	0	0	0
	02	sub	ra,1h				
->	04	jmp	2h				
	06	???					

Loop with unconditional jump

```
PC    Adr   Instr             RA    RB    RC    RD

      00    mov  ra,3h        0     0     0     0
->    02    sub  ra,1h
      04    jmp  2h
      06    ???
```

Loop with unconditional jump

```
PC    Adr   Instr              RA    RB    RC    RD

      00    mov   ra,3h        FF    0     0     0
      02    sub   ra,1h
->    04    jmp   2h
      06    ???
```

Loop with unconditional jump

PC	Adr	Instr	RA	RB	RC	RD
	00	mov ra,3h	FF	0	0	0
->	02	sub ra,1h				
	04	jmp 2h				
	06	???				

Loop with unconditional jump

PC	Adr	Instr		RA	RB	RC	RD
	00	mov	ra,3h	FE	0	0	0
	02	sub	ra,1h				
->	04	jmp	2h				
	06	???					

Loop with unconditional jump

PC	Adr	Instr	RA	RB	RC	RD
	00	mov ra,3h	FE	0	0	0
->	02	sub ra,1h				
	04	jmp 2h				
	06	???				

Loop with unconditional jump

```
PC    Adr   Instr              RA    RB    RC    RD

      00    mov   ra,3h        FE    0     0     0
->    02    sub   ra,1h
      04    jmp   2h
      06    ???
```

This line is never executed!

Conditional jumps

- Conditional jumps: changes the value of PC if a condition is met.
- Condition is tested by checking the flags (carry, zero).
- Flags are set by a prior comparison

- JA: jump if above
  - Jumps if Zero=0 and Carry=0
- JB: jump if below
  - Jumps if Zero=0 and Carry=1
- JE: jump if equal
  - Jumps if Zero=1

- And the opposite: JNA, JNB, JNE (not above, not below, not equal)

Loop with conditional jump (jne)

- Loop with variable from startvalue to 0 (inclusive)

Loop with conditional jump (jne)

```
PC    Adr   Instr             RA  RB  RC  RD    FLAGS
                              0   0   0   0

->    00    mov   ra,3h
      02    sub   ra,1h
      04    cmp   ra,0h
      06    jne   02
      08    ???
```

Loop with conditional jump (jne)

```
PC    Adr   Instr              RA  RB  RC  RD    FLAGS
                               3   0   0   0
      00    mov   ra,3h
->    02    sub   ra,1h
      04    cmp   ra,0h
      06    jne   02
      08    ???
```

Loop with conditional jump (jne)

```
PC    Adr   Instr            RA  RB  RC  RD    FLAGS
                             2   0   0   0
      00    mov  ra,3h
      02    sub  ra,1h
->    04    cmp  ra,0h
      06    jne  02
      08    ???
```

Loop with conditional jump (jne)

```
PC   Adr  Instr          RA RB RC RD   FLAGS
                         2  0  0  0
     00   mov  ra,3h
     02   sub  ra,1h
     04   cmp  ra,0h
->   06   jne  02
     08   ???
```

Loop with conditional jump (jne)

```
PC    Adr   Instr                 RA  RB  RC  RD    FLAGS
                                  2   0   0   0
      00    mov  ra,3h
->    02    sub  ra,1h
      04    cmp  ra,0h
      06    jne  02
      08    ???
```

Loop with conditional jump (jne)

```
PC    Adr   Instr          RA  RB  RC  RD    FLAGS
                           1   0   0   0
      00    mov   ra,3h
      02    sub   ra,1h
->    04    cmp   ra,0h
      06    jne   02
      08    ???
```

Loop with conditional jump (jne)

```
PC    Adr   Instr               RA  RB  RC  RD    FLAGS
                                1   0   0   0
      00    mov  ra,3h
      02    sub  ra,1h
      04    cmp  ra,0h
->    06    jne  02
      08    ???
```

Loop with conditional jump (jne)

```
PC    Adr   Instr             RA  RB  RC  RD    FLAGS
                              1   0   0   0
      00    mov   ra,3h
->    02    sub   ra,1h
      04    cmp   ra,0h
      06    jne   02
      08    ???
```

Loop with conditional jump (jne)

```
PC    Adr   Instr            RA  RB  RC  RD    FLAGS
                             0   0   0   0
      00    mov  ra,3h
      02    sub  ra,1h
->    04    cmp  ra,0h
      06    jne  02
      08    ???
```

Loop with conditional jump (jne)

```
PC    Adr   Instr              RA  RB  RC  RD    FLAGS
                               0   0   0   0    Z
      00    mov  ra,3h
      02    sub  ra,1h
      04    cmp  ra,0h
->    06    jne  02
      08    ???
```

Loop with conditional jump (jne)

```
PC    Adr   Instr                 RA  RB  RC  RD    FLAGS
                                  0   0   0   0    Z
      00    mov  ra,3h
      02    sub  ra,1h
      04    cmp  ra,0h
      06    jne  02
->    08    ???
```

Loop with conditional jump (jne)

```
PC    Adr   Instr               RA  RB  RC  RD    FLAGS
                                2   0   0   0    Z
      00    mov  ra,3h
      02    sub  ra,1h
      04    cmp  ra,0h
      06    jne  02
->    08    ???
```

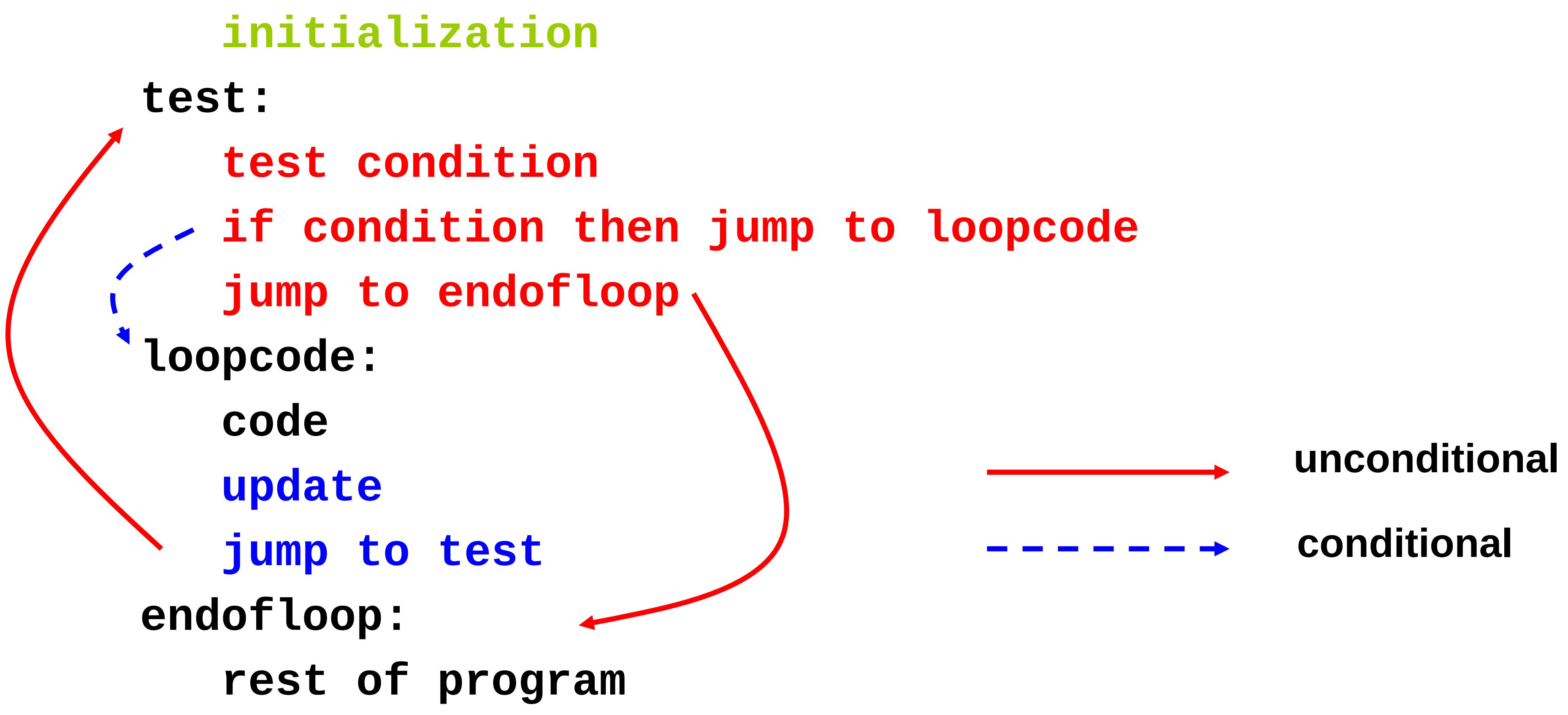
Program continues execution

C to assembler loops

- In C the syntax for a for loop is:
- for(<initialization>;<condition>; <update>) {code}
- Example: for(i=3; i != 0; i--) {....}
  - i will take the value: 3, 2, 1.
- Can be easily translated to assembler

C to assembler loops: alternative 1

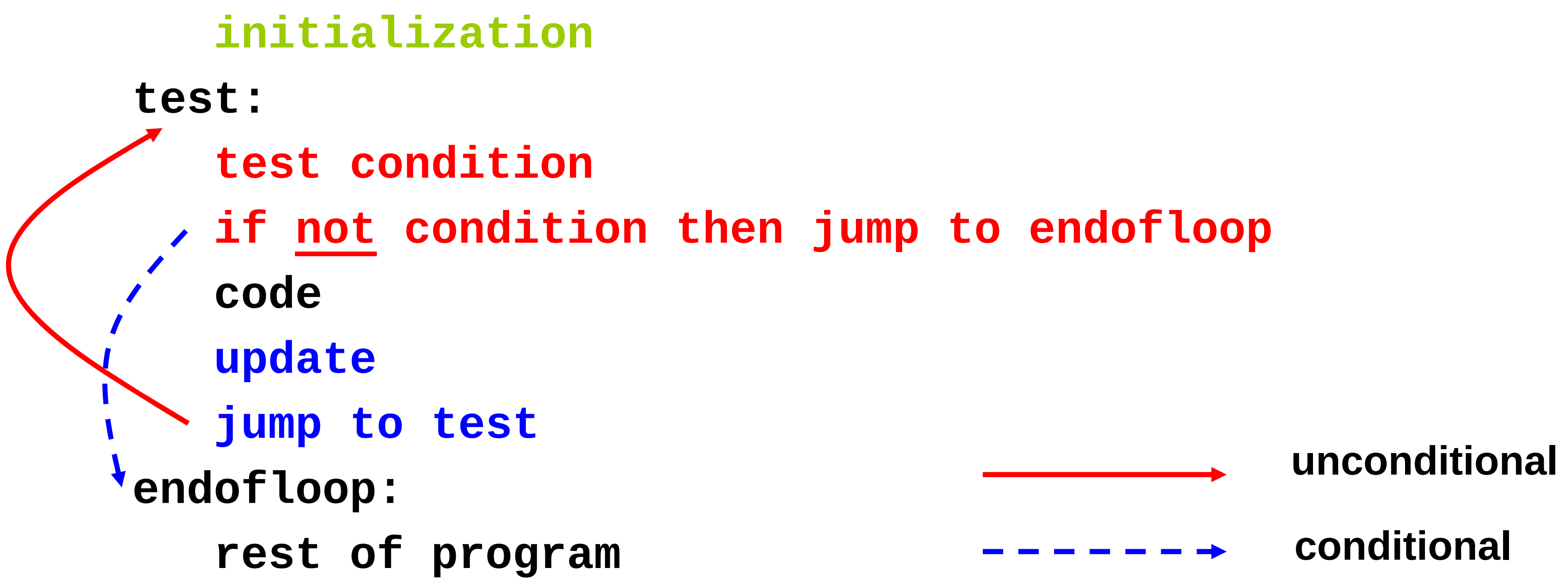
- for(<initialization>;<condition>; <update>) {code}



```
    initialization
test:
    test condition
    if condition then jump to loopcode
    jump to endofloop
loopcode:
    code
    update
    jump to test
endofloop:
    rest of program
```

C to assembler loops: alternative 2

- for(<initialization>;<condition>; <update>) {code}



```
    initialization
test:
    test condition
    if not condition then jump to endofloop
    code
    update
    jump to test
endofloop:
    rest of program
```

This requires to negate the condition!
Processors usually provide conditional jumps if condition (je,ja,jb)
and conditional jumps if not condition (jne,jna,jnb)

C to assembler loops

- With alternative 1 the conditional jump is to a nearby address (instruction skip)
- Alternative 2 leads to more compact code
- Some processors have "relative jumps" that allow to change PC by an offset
  - On Intel/AMD x86 the "short relative jump" allows to offset PC by up to -128 to +127 bytes
- What happens on x86 if the loop code is longer than 127 bytes?
  - Alternative 2 cannot be used with short relative jump!
  - Alternative 1 must be used

Programming influenced by processor architecture!
Higher level languages (e.g. C) and compilers allow to select the right assembler construct to optimize the code

C to assembler loops

- for(i=3; i != 0; i--) {....}

- Alternative 2

for(i=3; i != 0; i--) {....}

PC	Adr	Instr	RA	RB	RC	RD	FLAGS
			0	0	0	0	
->	00	mov ra,3h					
	02	cmp ra,0h					
	04	je 0eh					
	06	...					
	08	...					
	0A	sub ra,1h					
	0C	jmp 02					
	0E	???					

for(i=3; i != 0; i--) {....}

```
PC    Adr   Instr              RA RB RC RD    FLAGS
                               3  0  0  0
      00    mov  ra,3h
->    02    cmp  ra,0h
      04    je   0eh
      06    ...
      08    ...
      0A    sub  ra,1h
      0C    jmp  02
      0E    ???
```

for(i=3; i != 0; i--) {....}

```
PC    Adr   Instr            RA  RB  RC  RD    FLAGS
                             3   0   0   0
      00    mov  ra,3h
      02    cmp  ra,0h
->    04    je   0eh
      06    ...
      08    ...
      0A    sub  ra,1h
      0C    jmp  02
      0E    ???
```

for(i=3; i != 0; i--) {....}

```
PC    Adr   Instr               RA  RB  RC  RD    FLAGS
                                3   0   0   0
      00    mov   ra,3h
      02    cmp   ra,0h
      04    je    0eh
->    06    ...
      08    ...
      0A    sub   ra,1h
      0C    jmp   02
      0E    ???
```

for(i=3; i != 0; i--) {....}

```
PC    Adr   Instr               RA  RB  RC  RD    FLAGS
                                3   0   0   0
      00    mov   ra,3h
      02    cmp   ra,0h
      04    je    0eh
      06    ...
      08    ...
->    0A    sub   ra,1h
      0C    jmp   02
      0E    ???
```

for(i=3; i != 0; i--) {....}

```
PC    Adr   Instr              RA  RB  RC  RD    FLAGS
                               2   0   0   0
      00    mov   ra,3h
      02    cmp   ra,0h
      04    je    0eh
      06    ...
      08    ...
      0A    sub   ra,1h
->    0C    jmp   02
      0E    ???
```

for(i=3; i != 0; i--) {....}

```
PC    Adr   Instr                   RA  RB  RC  RD     FLAGS
                                    2   0   0   0
      00    mov   ra,3h
->    02    cmp   ra,0h
      04    je    0eh
      06    ...
      08    ...
      0A    sub   ra,1h
      0C    jmp   02
      0E    ???
```

for(i=3; i != 0; i--) {....}

```
PC     Adr    Instr                RA  RB  RC  RD     FLAGS
                                   2   0   0   0
       00     mov   ra,3h
       02     cmp   ra,0h
->     04     je    0eh
       06     ...
       08     ...
       0A     sub   ra,1h
       0C     jmp   02
       0E     ???
```

for(i=3; i != 0; i--) {....}

```
PC    Adr   Instr              RA  RB  RC  RD    FLAGS
                               2   0   0   0
      00    mov   ra,3h
      02    cmp   ra,0h
      04    je    0eh
->    06    ...
      08    ...
      0A    sub   ra,1h
      0C    jmp   02
      0E    ???
```

for(i=3; i != 0; i--) {....}

```
PC    Adr   Instr              RA  RB  RC  RD    FLAGS
                               2   0   0   0
      00    mov  ra,3h
      02    cmp  ra,0h
      04    je   0eh
      06    ...
      08    ...
->    0A    sub  ra,1h
      0C    jmp  02
      0E    ???
```

for(i=3; i != 0; i--) {....}

```
PC    Adr   Instr             RA RB RC RD    FLAGS
                              1  0  0  0
      00    mov  ra,3h
      02    cmp  ra,0h
      04    je   0eh
      06    ...
      08    ...
      0A    sub  ra,1h
->    0C    jmp  02
      0E    ???
```

for(i=3; i != 0; i--) {....}

```
PC    Adr   Instr            RA  RB  RC  RD    FLAGS
                             1   0   0   0
      00    mov  ra,3h
->    02    cmp  ra,0h
      04    je   0eh
      06    ...
      08    ...
      0A    sub  ra,1h
      0C    jmp  02
      0E    ???
```

for(i=3; i != 0; i--) {....}

```
PC    Adr   Instr               RA  RB  RC  RD    FLAGS
                                1   0   0   0
      00    mov  ra,3h
      02    cmp  ra,0h
->    04    je   0eh
      06    ...
      08    ...
      0A    sub  ra,1h
      0C    jmp  02
      0E    ???
```

for(i=3; i != 0; i--) {....}

```
PC     Adr   Instr              RA  RB  RC  RD    FLAGS
                                1   0   0   0
       00    mov  ra,3h
       02    cmp  ra,0h
       04    je   0eh
->     06    ...
       08    ...
       0A    sub  ra,1h
       0C    jmp  02
       0E    ???
```

for(i=3; i != 0; i--) {....}

```
PC    Adr   Instr              RA  RB  RC  RD    FLAGS
                               1   0   0   0
      00    mov  ra,3h
      02    cmp  ra,0h
      04    je   0eh
      06    ...
      08    ...
->    0A    sub  ra,1h
      0C    jmp  02
      0E    ???
```

for(i=3; i != 0; i--) {....}

```
PC     Adr   Instr                 RA  RB  RC  RD    FLAGS
                                   0   0   0   0
       00    mov   ra,3h
       02    cmp   ra,0h
       04    je    0eh
       06    ...
       08    ...
       0A    sub   ra,1h
->     0C    jmp   02
       0E    ???
```

for(i=3; i != 0; i--) {....}

```
PC     Adr   Instr               RA  RB  RC  RD    FLAGS
                                 0   0   0   0
       00    mov   ra,3h
->     02    cmp   ra,0h
       04    je    0eh
       06    ...
       08    ...
       0A    sub   ra,1h
       0C    jmp   02
       0E    ???
```

for(i=3; i != 0; i--) {....}

```
PC    Adr   Instr               RA  RB  RC  RD    FLAGS
                                0   0   0   0   Z
      00    mov  ra,3h
      02    cmp  ra,0h
->    04    je   0eh
      06    ...
      08    ...
      0A    sub  ra,1h
      0C    jmp  02
      0E    ???
```

for(i=3; i != 0; i--) {....}

```
PC    Adr   Instr             RA  RB  RC  RD    FLAGS
                              0   0   0   0   Z
      00    mov  ra,3h
      02    cmp  ra,0h
      04    je   0eh
      06    ...
      08    ...
      0A    sub  ra,1h
      0C    jmp  02
->    0E    ???
```

for(i=3; i != 0; i--) {....}

```
PC    Adr   Instr                 RA  RB  RC  RD    FLAGS
                                  0   0   0   0   Z
      00    mov   ra,3h
      02    cmp   ra,0h
      04    je    0eh
      06    ...
      08    ...
      0A    sub   ra,1h
      0C    jmp   02
->    0E    ???
```

Program continues execution

C to assembler loops

- for(i=3; i != 0; i--) {....}

- Alternative 1

```
->    00   mov   ra,3h
      02   cmp   ra,0h
      04   jne   08h
      06   jmp   0E
      08   ...
      0A   sub   ra,1h
      0C   jmp   02
      0E   ???
```

C to assembler loops

- for(i=0; i <3; i++) {....}
  - i takes the values: 0, 1, 2

- Alternative 1

for(i=0; i <3; i++) {....}

```
PC    Adr   Instr                RA  RB  RC  RD     FLAGS
                                 0   0   0   0
->    00    mov   ra,0h
      02    cmp   ra,3h
      04    jb    08h
      06    jmp   0E
      08    ...
      0A    add   ra,1h
      0C    jmp   02
      0E    ???
```

for(i=0; i <3; i++) {....}

```
PC    Adr   Instr           RA  RB  RC  RD    FLAGS
                            0   0   0   0
      00    mov   ra,0h
->    02    cmp   ra,3h
      04    jb    08h
      06    jmp   0E
      08    ...
      0A    add   ra,1h
      0C    jmp   02
      0E    ???
```

for(i=0; i <3; i++) {....}

```
PC    Adr   Instr                RA  RB  RC  RD    FLAGS
                                 0   0   0   0     C
      00    mov   ra,0h
      02    cmp   ra,3h
->    04    jb    08h
      06    jmp   0E
      08    ...
      0A    add   ra,1h
      0C    jmp   02
      0E    ???
```

for(i=0; i <3; i++) {....}

```
PC     Adr   Instr                RA  RB  RC  RD    FLAGS
                                  0   0   0   0     C
       00    mov   ra,0h
       02    cmp   ra,3h
       04    jb    08h
       06    jmp   0E
->     08    ...
       0A    add   ra,1h
       0C    jmp   02
       0E    ???
```

for(i=0; i <3; i++) {....}

```
PC    Adr   Instr             RA  RB  RC  RD    FLAGS
                              0   0   0   0     C
      00    mov   ra,0h
      02    cmp   ra,3h
      04    jb    08h
      06    jmp   0E
      08    ...
->    0A    add   ra,1h
      0C    jmp   02
      0E    ???
```

for(i=0; i <3; i++) {....}

```
PC     Adr   Instr                 RA  RB  RC  RD    FLAGS
                                   1   0   0   0     C
       00    mov   ra,0h
       02    cmp   ra,3h
       04    jb    08h
       06    jmp   0E
       08    ...
       0A    add   ra,1h
->     0C    jmp   02
       0E    ???
```

for(i=0; i <3; i++) {....}

```
PC    Adr   Instr                 RA  RB  RC  RD    FLAGS
                                  1   0   0   0     C
      00    mov   ra,0h
->    02    cmp   ra,3h
      04    jb    08h
      06    jmp   0E
      08    ...
      0A    add   ra,1h
      0C    jmp   02
      0E    ???
```

for(i=0; i <3; i++) {....}

```
PC     Adr    Instr                      RA  RB  RC  RD     FLAGS
                                         1   0       0      C
       00     mov   ra,0h
       02     cmp   ra,3h
->     04     jb    08h
       06     jmp   0E
       08     ...
       0A     ad      ,1h
       0C           02
       0       ??
```

fast forward some steps...

for(i=0; i <3; i++) {....}

```
PC    Adr   Instr              RA  RB  RC  RD    FLAGS
                               2   0   0   0     C
      00    mov   ra,0h
      02    cmp   ra,3h
      04    jb    08h
      06    jmp   0E
      08    ...
->    0A    add   ra,1h
      0C    jmp   02
      0E    ???
```

for(i=0; i <3; i++) {....}

```
PC    Adr   Instr                RA  RB  RC  RD    FLAGS
                                 3   0   0   0     C
      00    mov  ra,0h
      02    cmp  ra,3h
      04    jb   08h
      06    jmp  0E
      08    ...
      0A    add  ra,1h
->    0C    jmp  02
      0E    ???
```

for(i=0; i <3; i++) {....}

```
PC    Adr   Instr              RA  RB  RC  RD    FLAGS
                               3   0   0   0     C
      00    mov   ra,0h
->    02    cmp   ra,3h
      04    jb    08h
      06    jmp   0E
      08    ...
      0A    add   ra,1h
      0C    jmp   02
      0E    ???
```

for(i=0; i <3; i++) {....}

```
PC    Adr   Instr              RA  RB  RC  RD    FLAGS
                               3   0   0   0   Z
      00    mov   ra,0h
      02    cmp   ra,3h
->    04    jb    08h
      06    jmp   0E
      08    ...
      0A    add   ra,1h
      0C    jmp   02
      0E    ???
```

for(i=0; i <3; i++) {....}

```
PC    Adr   Instr              RA  RB  RC  RD    FLAGS
                               3   0   0   0    Z
      00    mov  ra,0h
      02    cmp  ra,3h
      04    jb   08h
->    06    jmp  0E
      08    ...
      0A    add  ra,1h
      0C    jmp  02
      0E    ???
```

for(i=0; i <3; i++) {....}

```
PC    Adr   Instr                 RA  RB  RC  RD    FLAGS
                                  3   0   0   0   Z
      00    mov  ra,0h
      02    cmp  ra,3h
      04    jb   08h
      06    jmp  0E
      08    ...
      0A    add  ra,1h
      0C    jmp  02
->    0E    ???
```

for(i=0; i <3; i++) {....}

```
PC    Adr   Instr               RA  RB  RC  RD    FLAGS
                                3   0   0   0   Z
      00    mov  ra,0h
      02    cmp  ra,3h
      04    jb   08h
      06    jmp  0E
      08    ...
      0A    add  ra,1h
      0C    jmp  02
->    0E    ???
```

Program continues execution

Summary

- Basic loop constructs can be realized in assembler
- Pay attention to the desired range of values of the variables and where the test is placed!
- The "C to assembler" examples generalize to more complex tests!
  - for(i=0; (i<100) && (obstacle1==0); i++) {....}
  - Use boolean logic to combine multiple simple tests together
  - Or test individual parts and have several conditional jumps
- Knowledge sufficient to complete the coursework assignment involving programming