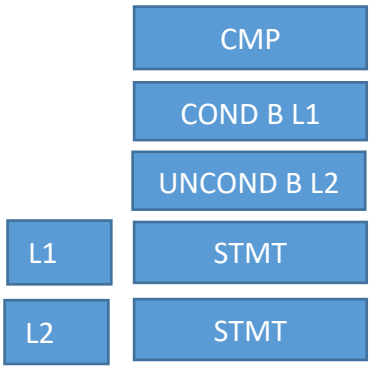
ADDITIONAL NOTES: ASSEMBLY LANGUAGE

- Assembly language is machine dependent. Language for Intel machine is different with language for Sun machine.
- Assembly language for **CSC569** follows the rules below:
  - There is **ONLY ONE** Register: R1
  - All arithmetic operations must be done using R1
- Possible operations are listed in Table below.

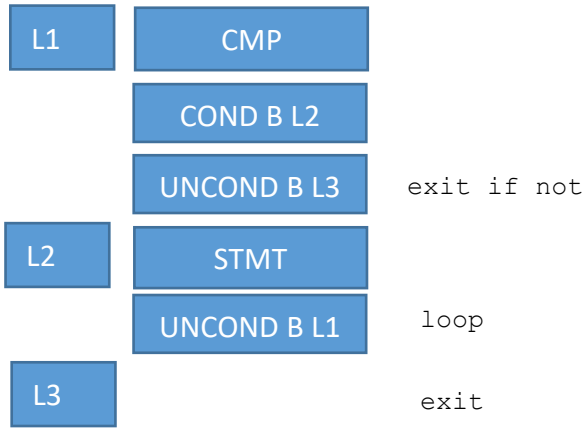
OPERATION	INFORMATION	EXAMPLE	NOTES
LOD	Load data into register	`LOD R1, a`	Must use with R1
STO	Store data from register to variable	`STO RI, T1`	Must use with R1
PUSH	Push value on top of stack	`PUSH a`	
CALL	Call function	`CALL println`	
CMP	Compare between two values	`CMP a, T1`	Cannot be used with R1 or numbers. Store into T1, T2 etc first.
ARITHMETIC OPERATION			
ADD	Add numbers	`ADD R1, b`	Must use with R1
SUB	Subtract numbers	`SUB R1, b`	Must use with R1
MUL	Multiply numbers	`MUL R1, b`	Must use with R1
DIV	Divide numbers	`DIV R1, b`	Must use with R1
BRANCH OPERATION			
B	Unconditional branch	`B L2`	
Conditional branch			Boolean operator
BH	Branch if higher	`BH L3`	>
BL	Branch if lower	`BL L3`	<
BHE	Branch if higher or even	`BHE L3`	>=
BE	Branch if equal	`B L2`	==
BNE	Branch if not equal	`BNE L2`	!=
BLE	Branch if lower or even	`BLE L3`	<=
L	Label a place	`L2`	

ASSEMBLY LANGUAGE FOR CONTROL STRUCTURES

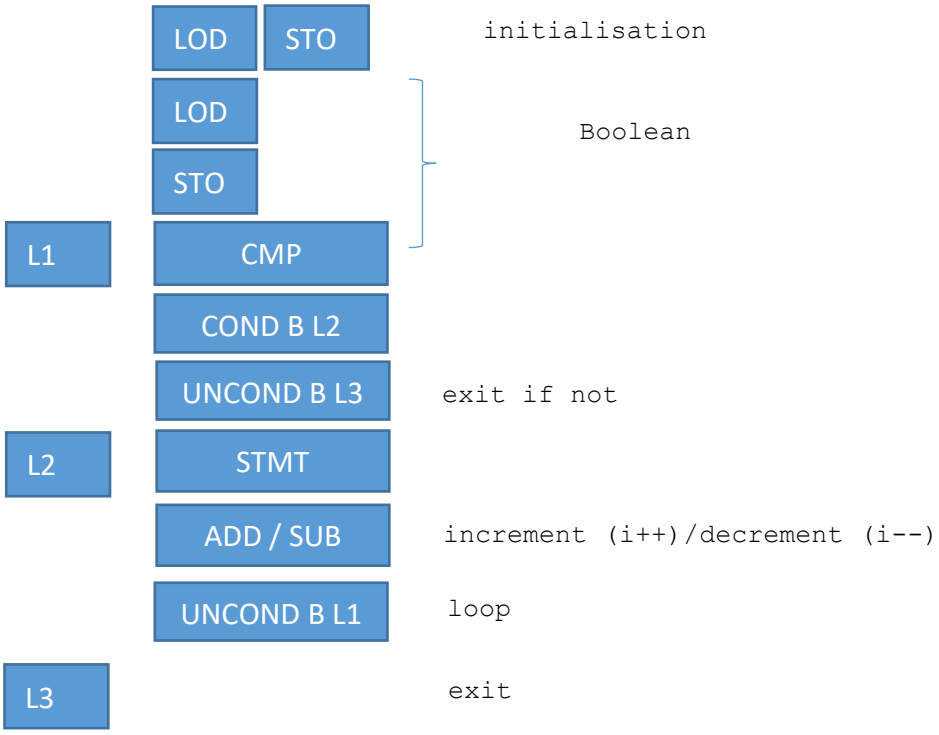
1. if-else



2. while



3. for



Examples of operations:

1. c = a + b;

```
LOD R1, a
ADD R1, b
STO R1, c
```

2. a = 14;
   b = 12;
   println(a-b);

```
LOD R1, = '14'
STO R1, a
LOD R1, = '12'
LOD R1, a
SUB R1, b
STO R1, T1
PUSH T1
CALL PRINTLN
```

3. a + b + c

```
LOD R1, a
ADD R1, b
ADD R1, c
STO R1, T1
```

4. while (x <= a+b) x = 2*x

```
L1:   LOD R1, a
      ADD R1, b
      STO R1, T1
      CMP x, T1
      BL L2
      B L3
L2:   LOD R1, = '2'
      MUL R1, x
      STO R1, x
      B L1
L3:
```

5. if (a > b) a = a+1

```
      CMP a, b
      BH L1
      B L2
L1:   LOD R1, a
      ADD R1, 1
      STO R1, a
L2:
```

6. for (a=1; a<=20; a++)
   y = y + 1

```
      LOD R1, '='
      STO R1, a
      LOD R1, ='20'
      STO R1, T1
L1:   CMP a, T1
      BLE L2
      B L3
L2:   LOD R1, y
      ADD R1, ='1'
      STO R1, y
      LOD R1, a
      ADD R1, ='1'
      STO R1, a
      B L1
L3:
```