

Computer Science CSCI 261

Computer Architecture and Assembly Language

*Dr. Peter Walsh
Department of Computer Science
Vancouver Island University
peter.walsh@viu.ca*

SSBC Program Development

- Step Wise Refinement (Variation)
 - develop a HLL (or psedo code) solution
 - verify behaviour (testing and inspection)

 - convert HLL to assembly language
 - verify behaviour (inspection)

 - convert assembly language to machine language
 - verify behaviour (testing and inspection)

Sigma 5

○ $\Sigma 5 = 5 + 4 + 3 + 2 + 1 = 15$

```
#include <stdio.h>
```

```
int main (void) {
```

```
    int sum = 0;
```

```
    int n = 5;
```

```
    L: sum = sum + n;
```

```
    n = n - 1;
```

```
    printf("sum = %i \t n = %i \n", sum, n);
```

```
    if (n != 0) goto L;
```

```
    return (0);
```

```
}
```

Sigma 5 cont.

○ Trace

sum = 5 n = 4

sum = 9 n = 3

sum = 12 n = 2

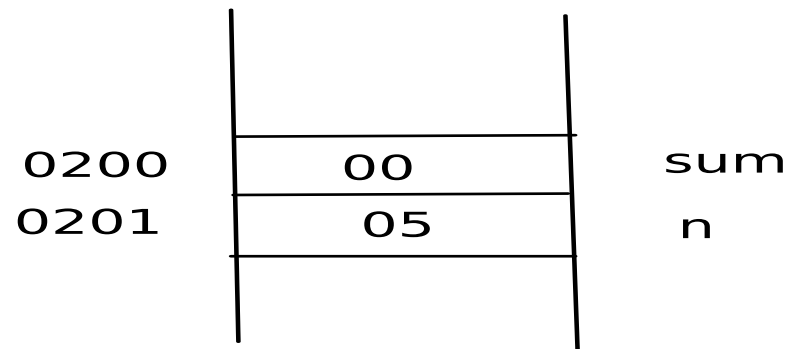
sum = 14 n = 1

sum = 15 n = 0

- test and inspect for off-by-one errors in loops

SSBC Variables

- Global Variables
 - programmer reserves and manages space in memory
e.g. ($\Sigma 5$)



- Local Variables
 - programmer reserves and manages space
on the stack