

Basic programming structures in C and HC11



C to HC11 conversions

- addition
- increment
- decrement
- multiplication
- division
- **if**
- **if-else**
- **while**
- **do-while**
- arrays



C-HC11 conversions: addition

In C:

```
result = v0 + v1;
```

In HC11:

```
// 16-bit variables  
ldd v0  
add v1  
std result
```

```
// 8-bit variables  
ldab v0  
addb v1  
stab result
```

```
// mixed variables  
ldab v0  
abx // x += b
```



C-HC11 conversions: increment

In C:

```
++i;
```

In HC11:

```
// 16-bit variables  
inx // ++x  
iny // ++y  
ins // ++sp  
inc var16 // ++var16
```

```
// 8-bit variables  
inca // ++a  
incb // ++b  
inc var8 // ++var8
```



C-HC11 conversions: decrement

In C:

```
--i;
```

In HC11:

```
// 16-bit variables  
dex          // --x  
dey          // --y  
des          // --sp
```

```
// 8-bit variables  
deca         // --a  
decb         // --b  
dec 4, X     // --M[X+4]
```



"C" Conversions

C-HC11 conversions: multiplication

In C

```
int Result;          // 16-bit  
char Val0, Val1     // 8-bit  
  
Result = Val0 * Val1;
```

NOTE: 8-bit x 8-bit = 16-bit

A

B

D

In HC11:

```
ldaa Val0  
ldab Val1  
mul          // unsigned  
std Result
```



C-HC11 conversions: division

In C

```
int Quo, Rem; // 16-bit
int Val0, Val1; // 16-bit
Q = Val0 / Val1;
R = Val0 % Val1;
```

NOTE: 16-bit/16-bit = 16-bit, 16-bit

D

X

D

X

In HC11:

```
ldd Val0
ldx Val1
idiv // integer
stx Quo
std Rem
```



C-HC11 conversions: if

In C:

```
if(a > 10)
    y++;
```

In HC11:

```
// using ACCA and an immediate, both signed

    cmpa #10
    ble endif // if (a <= 10)
    iny
endif:
```



C-HC11 conversions: if-else

In C:

```
char a; // 8-bit
int y; // 16-bits

if(a > 10)
    y++;
else
    y--;
```

In HC11:

```
        cmpa #10
        bgt  if
        dey
        bra  endif
if:
        iny
endif:
```



C-HC11 conversions: while

In C:

```
char a, b;
while(a > 20)
{
    a--;
    b *= 3;
}
```

In HC11:

```
// Using Acc A and B
while:
        cmpa #20
        ble endwhile
        deca
        psha
        ldaa #3
        mul
        pula
        bra  while
endwhile:
```



C-HC11 conversions: do-while

In HC11:

In C:

```
int foo, bar, qux;
do
{
    foo -= 3;
    bar += qux;
} while (foo > 7)
```

```
DO:    ldd    foo
        subd   #3
        std    foo
        ldd    bar
        addd   qux
        std    bar
        ldd    foo
        cpd    #7
        bgt    DO
```



C-HC11 conv.: array sum

In C:

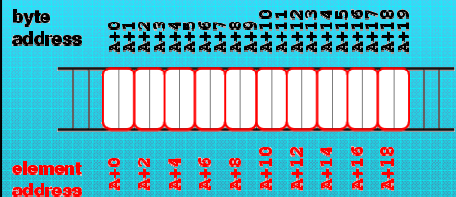
```
int    i = 0;
int    sum = 0;
int    A[10];

while(i < 10)
    sum += A[i++];
```

```
A:      .space   20 // int A[10];
        ldx    #0   // X = sum = 0;
        ldab   #0   // B = i = 0;

WHILE:  cmpb   #10  // if(i < 10)
        bgt    endwhile
        ldy    #array
        aby           // Y = &A[i];
        xgdx           // X ⇄ D
        addd   0, y   // D += A[i];
        xgdx           // D ⇄ X
        incb           // ++i
        incb           // (b += 2)
        bra    while

endwhile:
```



In HC11:

