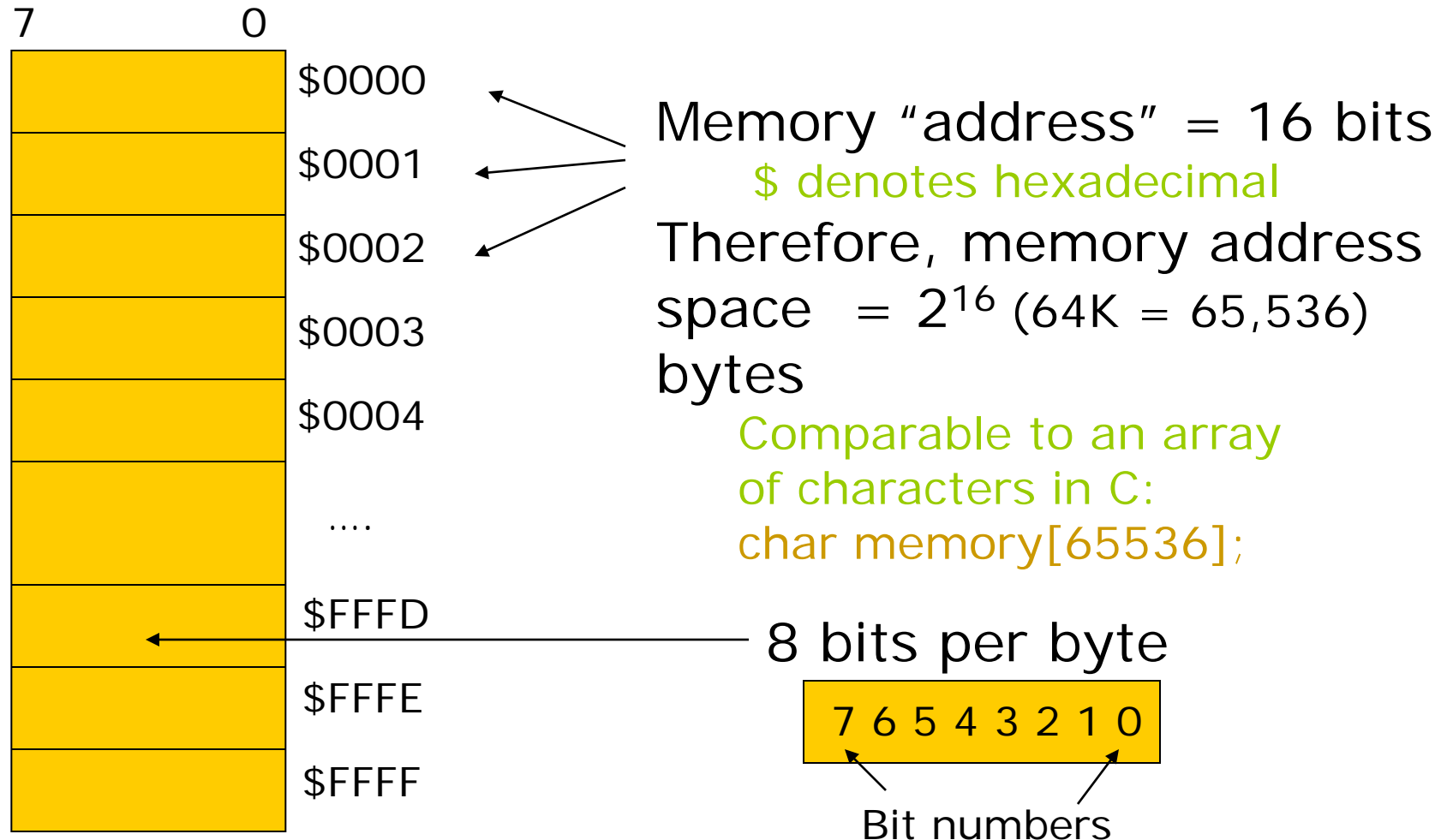


Chapter 5

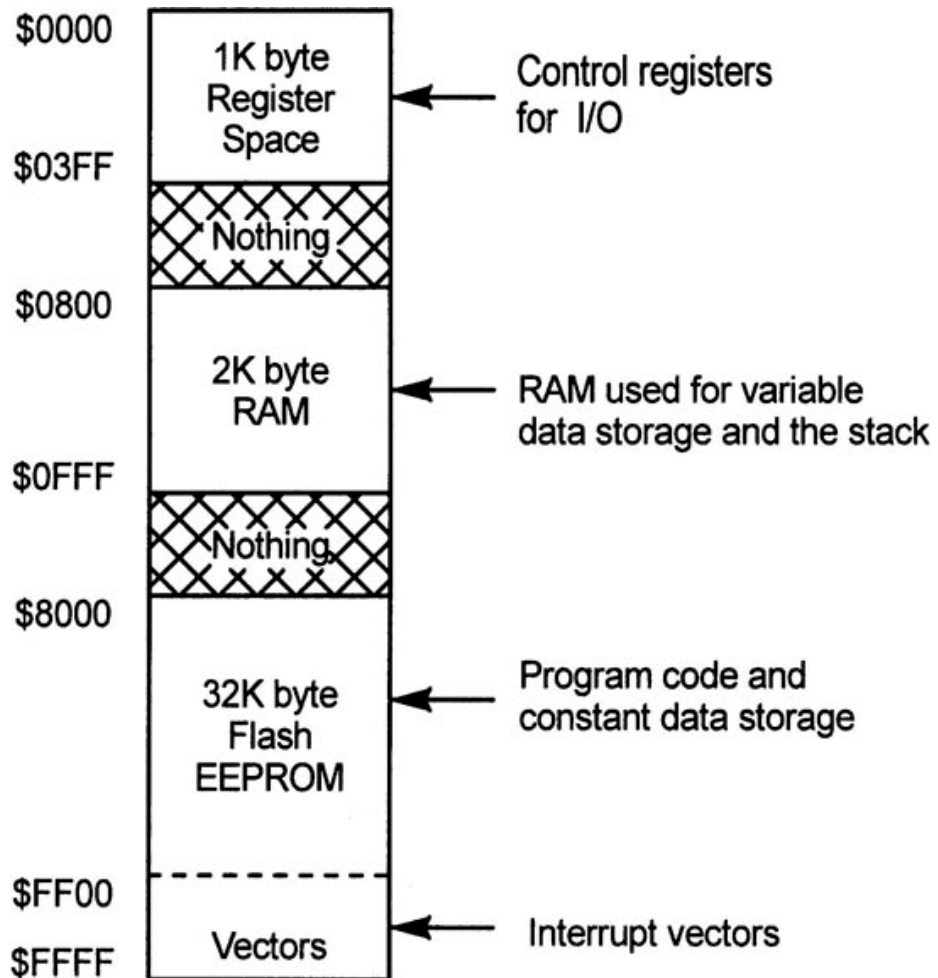


Memory Organization and
Assembler “Pseudoinstructions”
for setting up data and
constants in memory

HCS12 memory



MC9S12C32 memory map (Fig. 4.3)



On-chip
memory
addresses
are "fixed"

Designating available memory areas for MCS9C32 (in *Code Warrior*)

```
RAMStart EQU $0800 ; absolute address to place variables
RAMEnd EQU $0FFF ; last address of RAM, to init stack pointer
ROMStart EQU $8000 ; absolute address for code/constant data
```

```
; variable/data section
    ORG RAMStart
; Insert here your data definitions.
```

```
Counter DS.W 1
FiboRes DS.W 1
```

```
; code section
```

```
    ORG ROMStart
```

```
Entry:
```

```
    LDS #RAMEnd+1 ; initialize the stack pointer
;    other instructions here
```

```
ORG $FFFE
    DC.W Entry ; Reset Vector
```

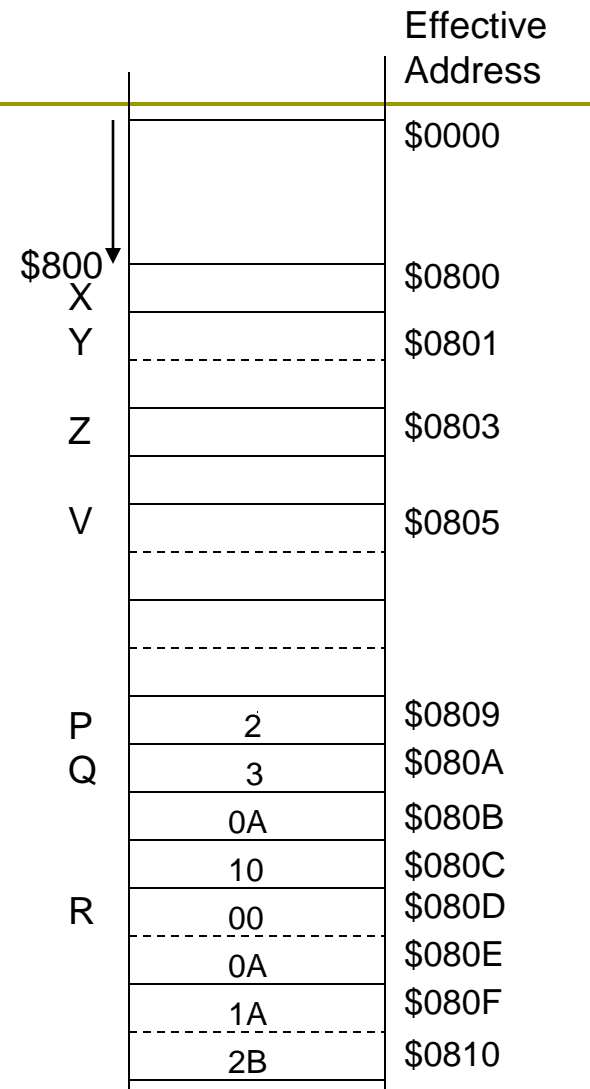
Assembler “Directives”

- Used to define variables, arrays, etc.
- These aren't instructions (not executable)

```
Const1 EQU $100
Const2 EQU $200
```

```
ORG $800
X DS.B 1 ; 1 byte X
Y DS.W 1 ; 2 bytes X
Z DS.B 2 ; 2 bytes
V DS.W 2 ; 4 bytes

P DC.B 2 ; a value of 2
Q DC.B 3,10,$10 ; values of 3, 10, $10
R DC.W 10,$1A2B ; values of 10, $1A2B
```



Constant values

- Constants can be in base 10, 2, 16, etc.
 - Decimal format is default: 256
 - Specify hexadecimal by leading \$: \$100 = 256
 - Specify binary by leading %: %100000000 = 256
- A constant can be “equated” to (represented by) a symbol for convenience

RAMstart equ \$0800

.....

org RAMstart ;same as writing “org \$0800”

↑
Semicolon designates a “comment”

Symbolic labels

- Data and program locations can be “labeled” for easy reference

```
RAMstart equ $0800
```

```
Counter: ds.w 1 ;allocate one memory word
```

```
.....
```

```
    Ldx Counter ;load IX register with  
                    ;value variable “Counter”
```

```
Same as: Ldx $0800
```

(“Counter” refers to address \$0800 in memory.)

Defining variables in memory

- ❑ Memory space must be allocated for each data value
- ❑ A symbolic label may be used to designate the location of each value

```
Bob:          ds.b 1          ;allocate 1 byte
Jim:          ds.w 1          ;allocate 1 word (2 bytes)
Carray:       ds.w 10         ;allocate 10 words (20 bytes)
Const1:       dc.b 15         ;allocate 1 byte & also
               ;store the value 15 in it
Const2:       dc.w 1500       ;allocate 1 word (2 bytes) &
               ;store the value 1500 in it
```

Assembler directives DS and DC

□ DS = “define storage”

- Allocate storage, leaving it uninitialized
- Format: `label: DS.B N`
 - Label (optional) is symbolic name for this address
 - .B indicates “byte” (.W for “word”)
 - N = # bytes to allocate

□ DC = “define constant storage”

- Allocate one or more bytes/words of storage and initialize to a designated values
- Format: `label: DC.B value1,value2,....`
 - Also DC.W for 2-byte “words” (values in “big-endian” format)