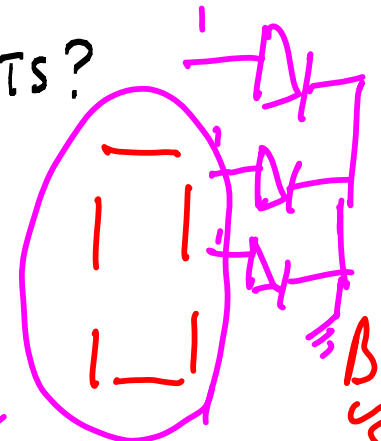
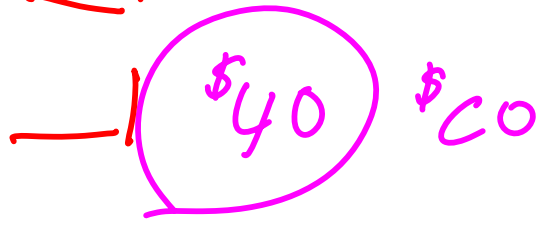
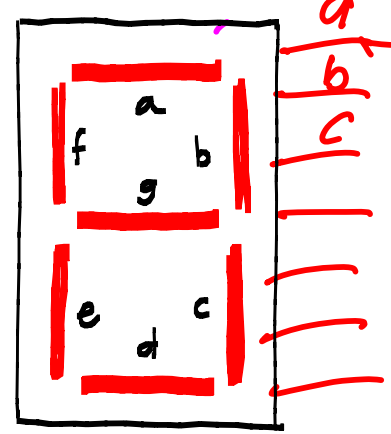


# HOW DO WE TRANSLATE OUR BITS INTO 7-SEG DIGITS?

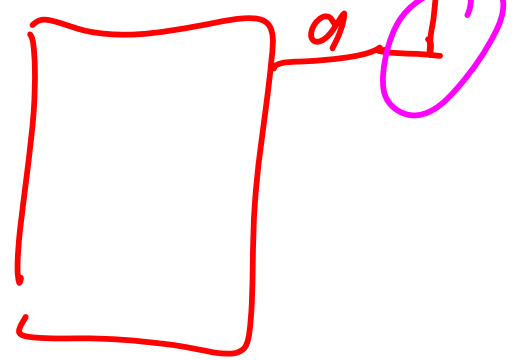


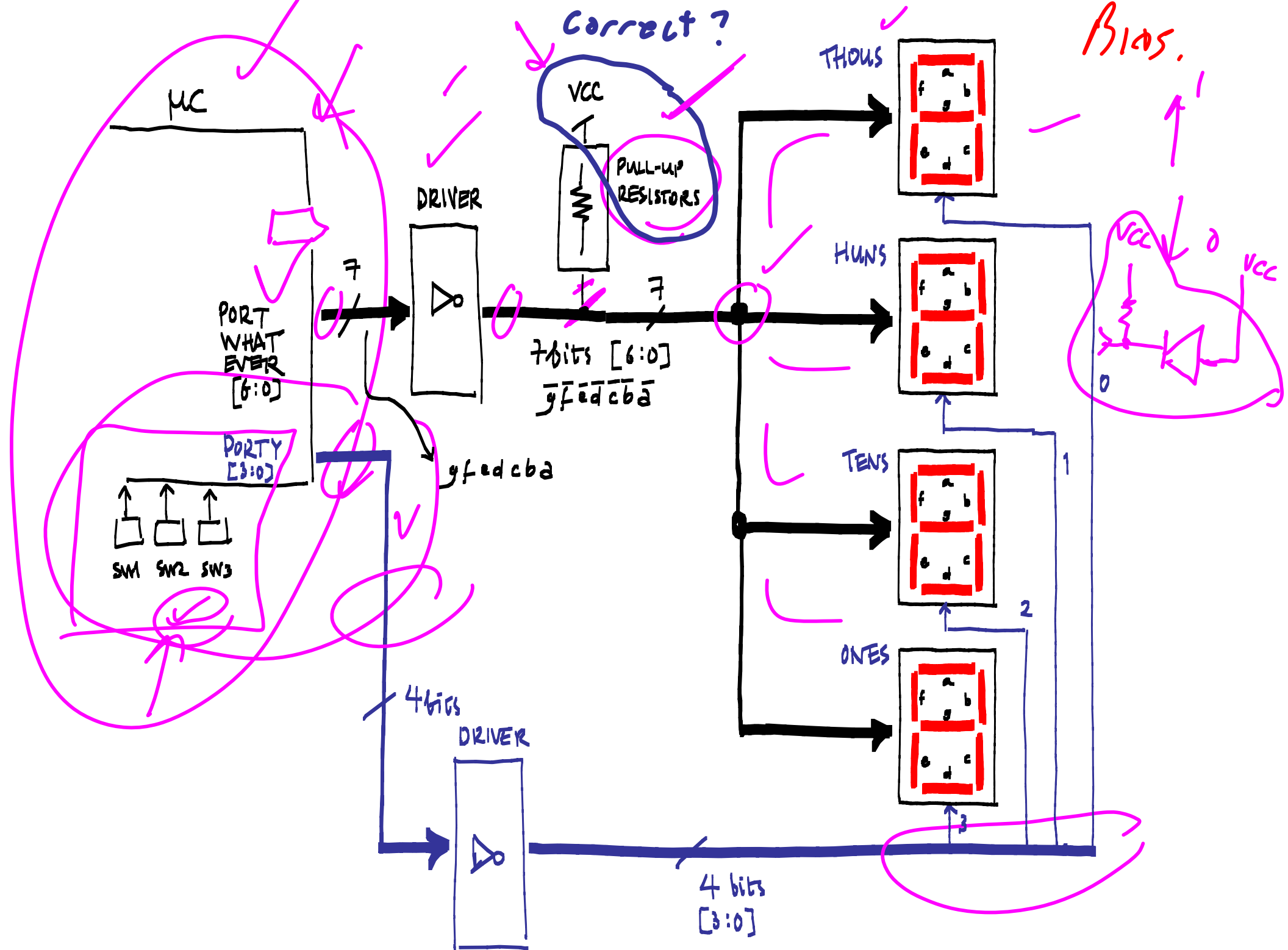
we build a correspondance table (depends on wiring !!)

| Number | Output of µC<br>Should be... | Segments ON<br>g f e d c b a | HEX WORD   |
|--------|------------------------------|------------------------------|------------|
| 0      | X 1 0 0 0 0 0 0              | X 0 1 1 1 1 1 1              | \$3F, \$BF |
| 1      |                              |                              |            |
| 2      |                              |                              |            |
| 3      |                              | X 1 0 0 1 1 1 1              | \$4F, \$CF |
| 4      |                              |                              |            |
| 5      |                              |                              |            |
| 6      |                              |                              |            |
| 7      |                              |                              |            |
| 8      |                              |                              |            |
| 9      |                              |                              |            |



Once we finish the table, what do we do with it?





Correct?

Bias.

MC

PORT WHAT EVER [6:0]

PORTY [3:0]

SW1 SW2 SW3

DRIVER

VCC

PULL-UP RESISTORS

7 bits [6:0]  
g f e d c b a

g f e d c b a

DRIVER

4 bits [3:0]

THOUS

HUNS

TENS

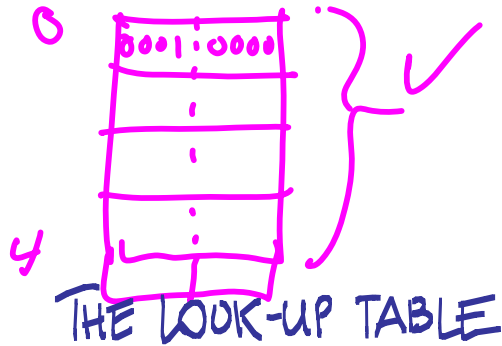
ONES



Now that we built a look-up table, how do we look it up?

let's see what it means to build it first

0 → 255



UNSIGNED char nums[10]  
 "8 bits each"

→ = { { 0x4d, 0x17, ... } ;  
 { 0000 0000 }

nums[0] = 0x4d  
 nums[1] = 0x17  
 ⋮

DECLARING VARIABLES

Signed chr

-128 → 127 (1 byte)  
 0 → 65535 (2)

Un signed int  
 int

-32768 → 32767 (2)

Un signed long int

0 → 46 (4)

long int

-26 → 26 (4)

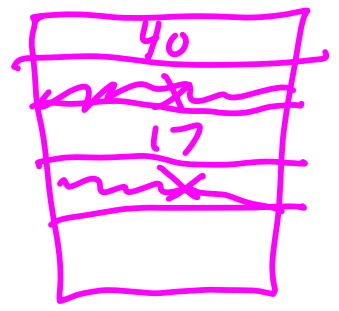
float

$1.18 \times 10^{-38} \rightarrow 3.4 \times 10^{+38}$  (4)

double

$2.23 \times 10^{-308} \rightarrow 1.8 \times 10^{+308}$  (8)

int nums[10]; = 0x40





# Hardware testing?



## Software functional blocks

- LUT ✓
- Switch decoding ✓
- Display update (Duty cycle) ✓
- Logic blocks (Events) ✓

