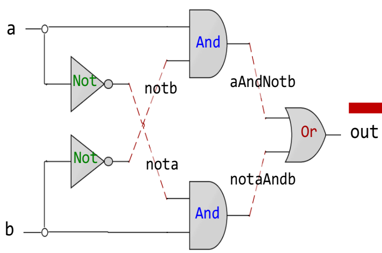
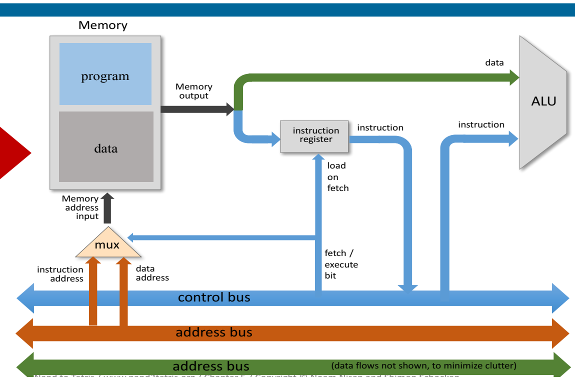
Computer Organization & Assembly Language Programming




```
CHIP Xor {
  IN a, b;
  OUT out;
  PARTS:
  Not(in=a, out=nota);
  Not(in=b, out=notb);
  And(a=nota, b=b, out=w1);
  And(a=a, b=notb, out=w2);
  Or(a=w1, b=w2, out=out);
}
```




```
@R1
D=M
@temp
M=D
```

```
0000000000000001
1111110000010000
0000000000010000
1110001100001000
```

Lecture # 18

Interfacing I/O Devices

```
#include<stdio.h>
#include<stdlib.h>
int main(){
  printf("Learning is fun with Arif\n");
  exit(0);
}
```

```
global main
SECTION .data
   msg: db "Learning is fun with Arif", 0Ah, 0h
   len_msg: equ $ - msg
SECTION .text
   main:
      mov rax,1
      mov rdi,1
      mov rsi,msg
      mov rdx,len_msg
      syscall
      mov rax,60
      mov rdi,0
      syscall
```

```
0:  b8 01 00 00 00
5:  bf 01 00 00 00
a:  48 be 00 00 00 00 00
11: 00 00 00
14: ba 1b 00 00 00
19: 0f 05
1b: b8 3c 00 00 00
20: bf 00 00 00 00
25: 0f 05
```

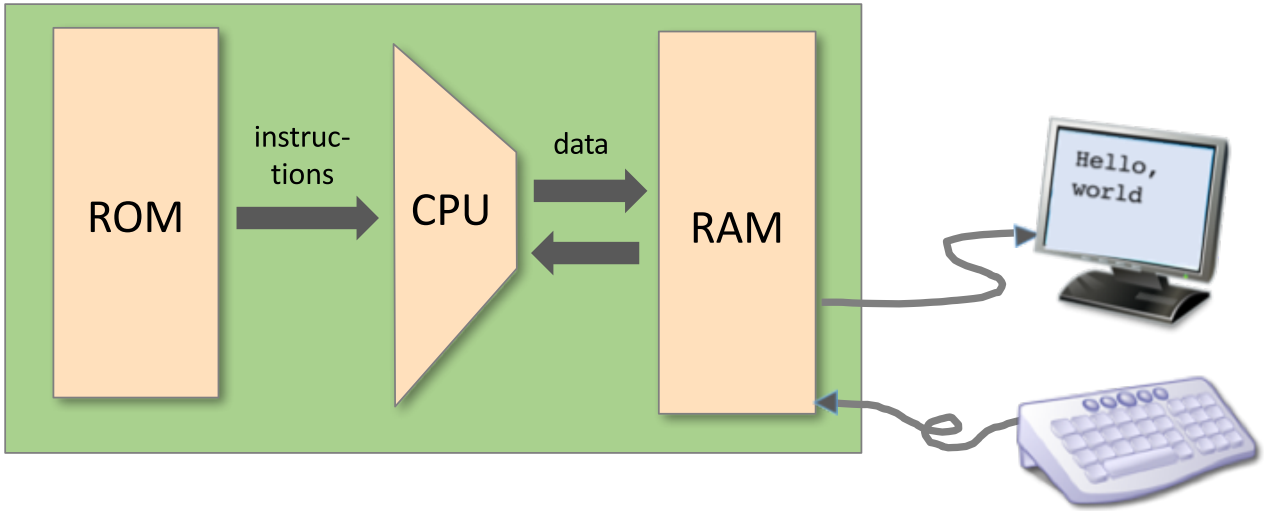
Slides of first half of the course are adapted from:
https://www.nand2tetris.org
Download s/w tools required for first half of the course from the following link:
https://drive.google.com/file/d/0B9c0BdDJz6XpZUh3X2dPR1o0MUE/view

Instructor: Muhammad Arif Butt, Ph.D.

Today's Agenda

- How to interface I/O devices with computer
- Interfacing Screen with Hack computer
  - Demo of built-in Screen chip on h/w Simulator
- Interfacing Keyboard with Hack computer
  - Demo of built-in Keyboard chip on h/w Simulator

Input / Output



Screen: used to display outputs

Keyboard: used to get data from user

I/O Handling

- **High Level Approach:** Sophisticated software library functions are used to display text/graphics on the monitor, read the keyboard, read voice notes from mic and play the audio on speakers etc
- **Low Level:** Bits Manipulation

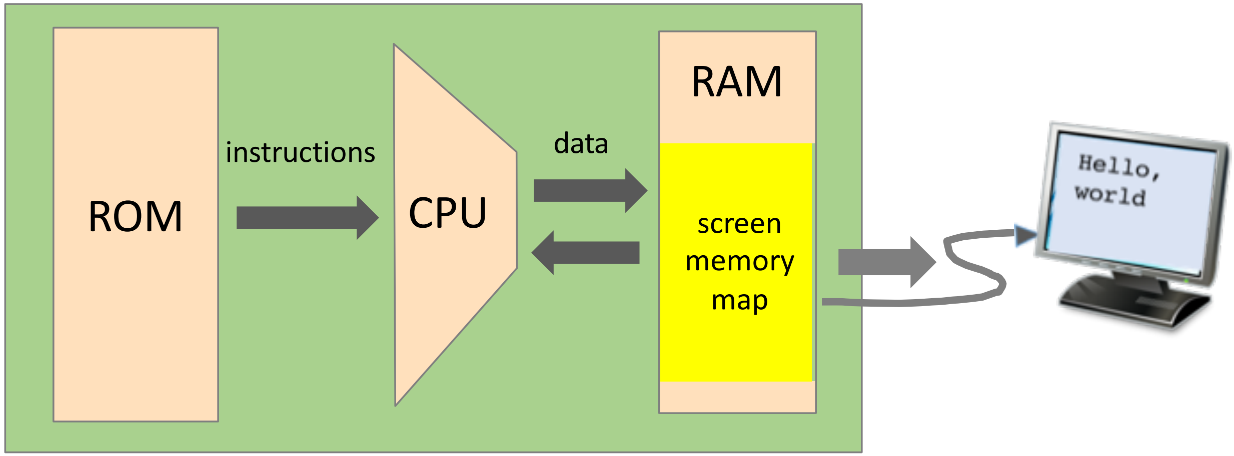
Interfacing I/O Devices with a Computer

- The way a microprocessor need to read/write different memory locations, similarly the microprocessor also need to read/write different I/O devices like the keyboard, mouse, monitor, printer, etc. This linking is also be called I/O Interfacing. An I/O interface acts as a communication channel between the processor and the externally interfaced device. The interfacing of the I/O devices can be done in two ways
  - **Memory Mapped I/O Interfacing:** Both memory and I/O devices have same address space. So addressing capability of memory become less because some part is occupied by the I/O. In memory mapped I/O, there are same read-write instructions for memory and I/O devices, so CPUs are cheaper, faster and easier to build
  - **Isolated I/O Interfacing:** The I/O devices are given a separate addressing region (separate from the memory). These separate address spaces are known as ‘Ports’. In isolated I/O, there are different read-write instructions for memory and I/O devices. x86-64 use Isolated I/O

Note: Data can be transferred between CPU and I/O devices in three modes, namely Program controlled I/O, Interrupt initiated I/O, and Direct Memory Access

Interfacing Screen with Hack Computer

Memory Mapped Output



Screen Memory Map:

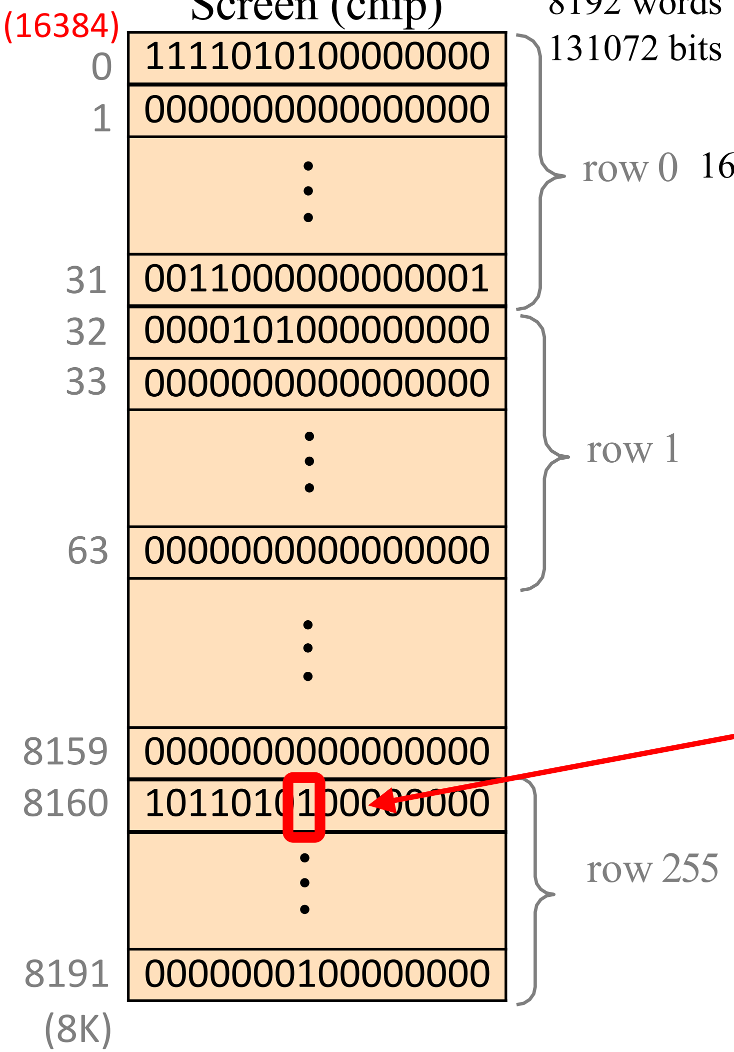
- Screen memory map is a designated memory area, dedicated to manage a display unit
- To write something on the display unit, write some bits in the designated memory area (zero to make a pixel off/white and one to make a pixel on/black)
- The physical display is continuously *refreshed* from the contents of memory map, many times per second
- Whatever, we write in the memory map makes the corresponding pixels of screen black and white in the next refresh cycle
- This is how we can write "Hello World" message on the screen

Screen Memory Map

Screen Memory Map

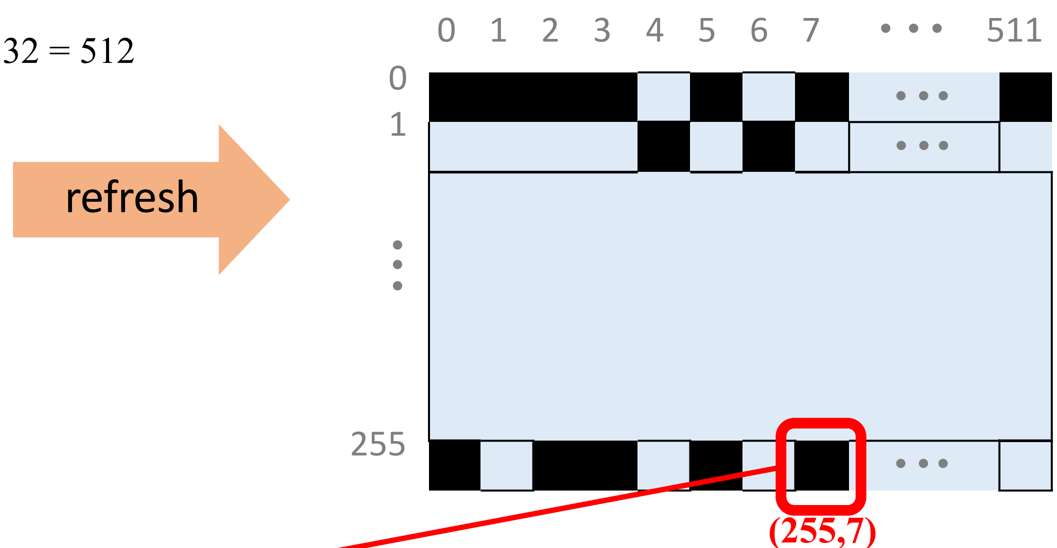
Memory Map
Screen (chip)

A sequence of 8K x 16 bit words
8192 words
131072 bits



Black & White Display Unit

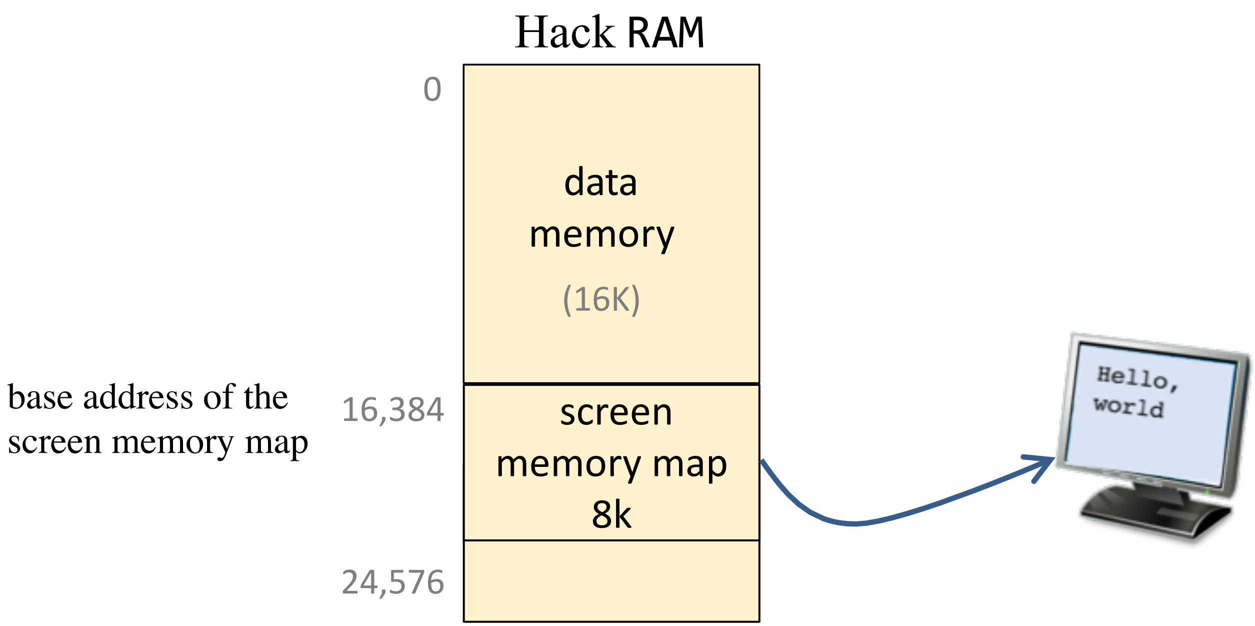
- A matrix of 256 rows x 512 columns
- 131072 pixels



To set pixel (row,col) on/off

```
word = Screen[32*row + col/16]
word = RAM[16384 + 32*row + col/16]
Set (col%16)th bit of word to 0 or 1
RAM[i] = word
```

Output

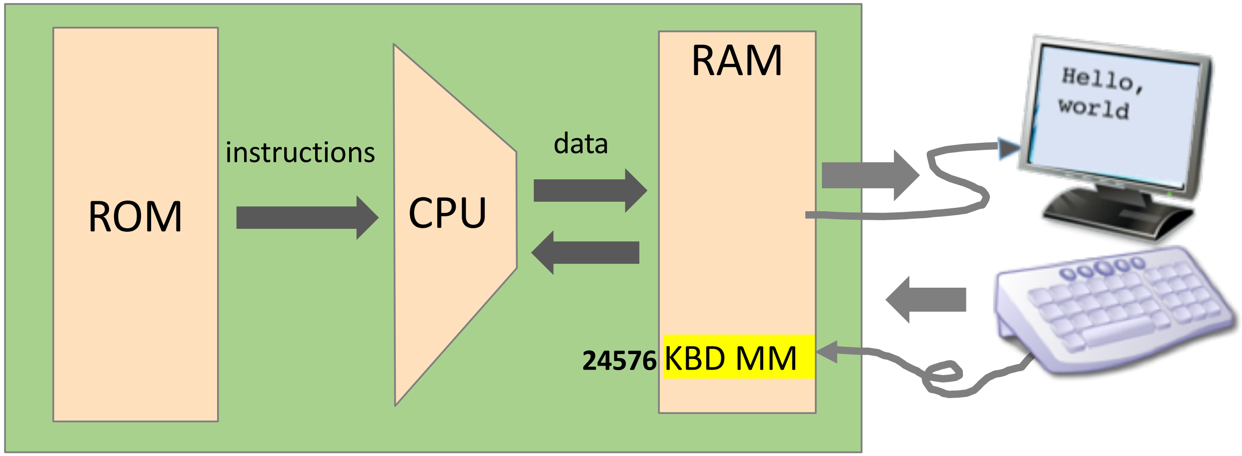


- The physical screen is of 256 rows and 512 columns which makes 256 x 512 = 131072 pixels
- To map each pixel of screen on a single bit, the Screen memory map must contain 8K, 16 bits words, which makes 8192 x 16 = 131072 bits

Screen Output Demo

Interfacing Keyboard with Hack Computer

Memory Mapped Input



<u>Keyboard Memory Map:</u>

- The physical keyboard is associated with a keyboard memory map, which is a designated RAM area, dedicated to manage the key board
- The physical screen was of 256 rows and 512 columns and the Screen memory map was of 131072 bits
- The Hack character set we need are less than 256, so for the keyboard we just need 16 bits, so the keyboard memory map is a single register at RAM address 24576

The Hacker Character Set

key	code
(space)	32
!	33
“	34
#	35
$	36
%	37
&	38
‘	39
(	40
)	41
*	42
+	43
,	44
-	45
.	46
/	47

key	code
0	48
1	49
…	…
9	57

:	58
;	59
<	60
=	61
>	62
?	63
@	64

key	code
A	65
B	66
C	…
…	…
Z	90

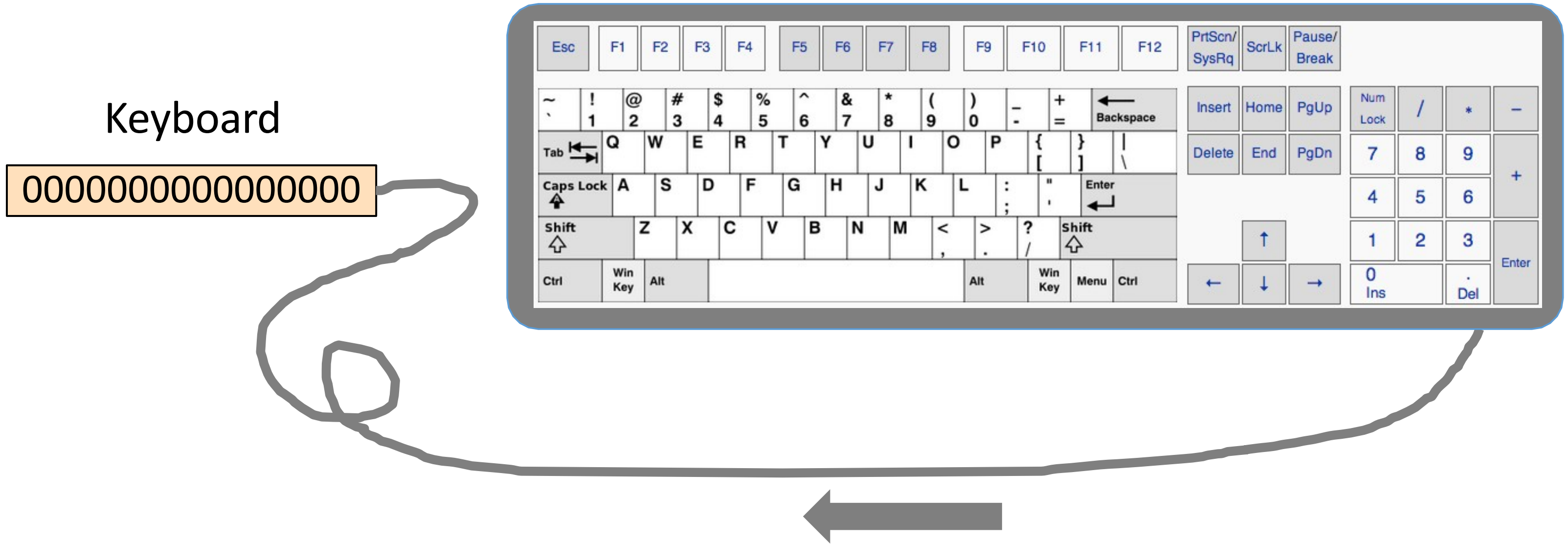
[	91
/	92
]	93
^	94
_	95
`	96

key	code
a	97
b	98
c	99
…	…
z	122

{	123
|	124
}	125
~	126

key	code
newline	128
backspace	129
left arrow	130
up arrow	131
right arrow	132
down arrow	133
home	134
end	135
Page up	136
Page down	137
insert	138
delete	139
esc	140
f1	141
…	…
f12	152

Memory Mapped Input

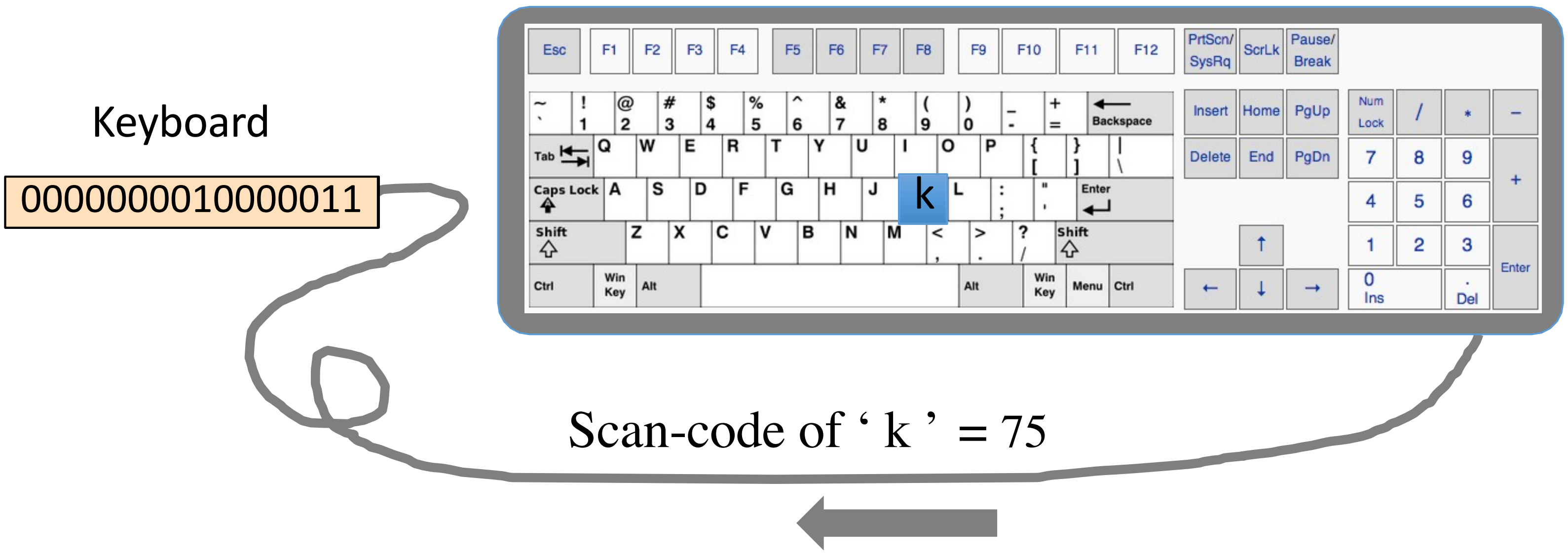


When a key is pressed on the keyboard, the key’s scan code appears in the keyboard memory map. Since no key is being pressed on the keyboard in this figure, so the keyboard memory map contains all zeros

<u>To check which key is currently pressed:</u>

- Probe the contents of the Keyboard chip
- In the Hack computer: probe the contents of RAM[24576]

Memory Mapped Input

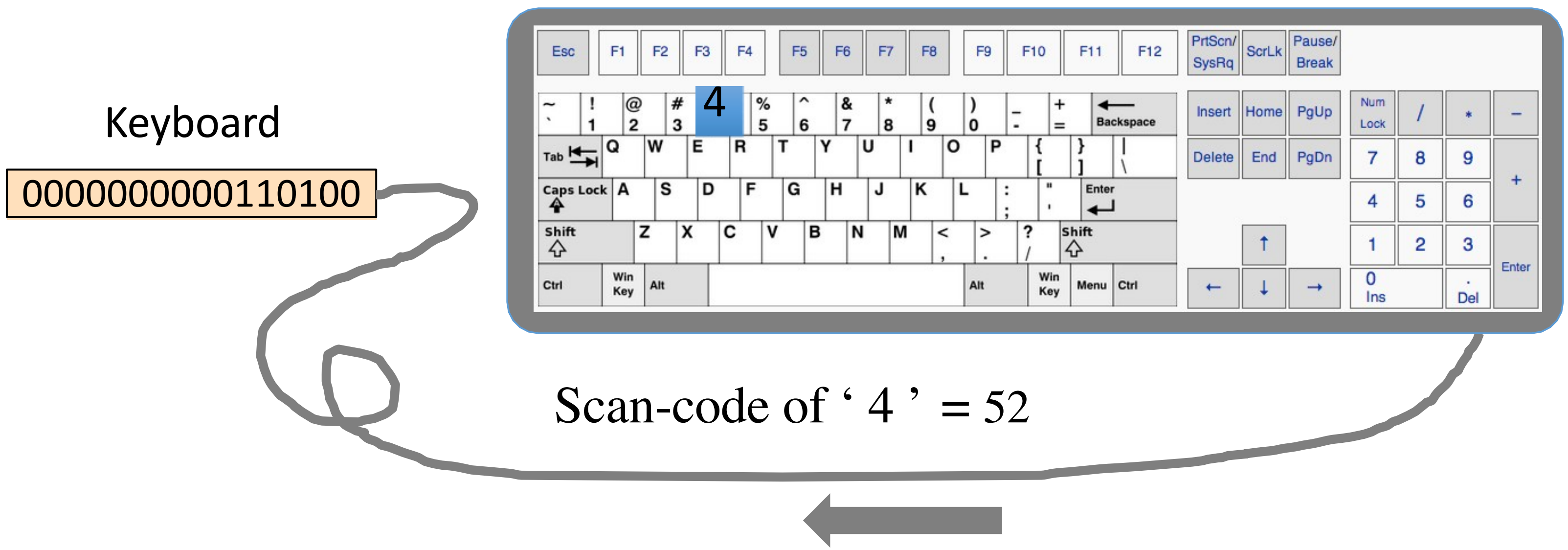


When a key is pressed on the keyboard, the key’s scan code appears in the keyboard memory map

<u>To check which key is currently pressed:</u>

- Probe the contents of the Keyboard chip
- In the Hack computer: probe the contents of RAM[24576]

Memory Mapped Input



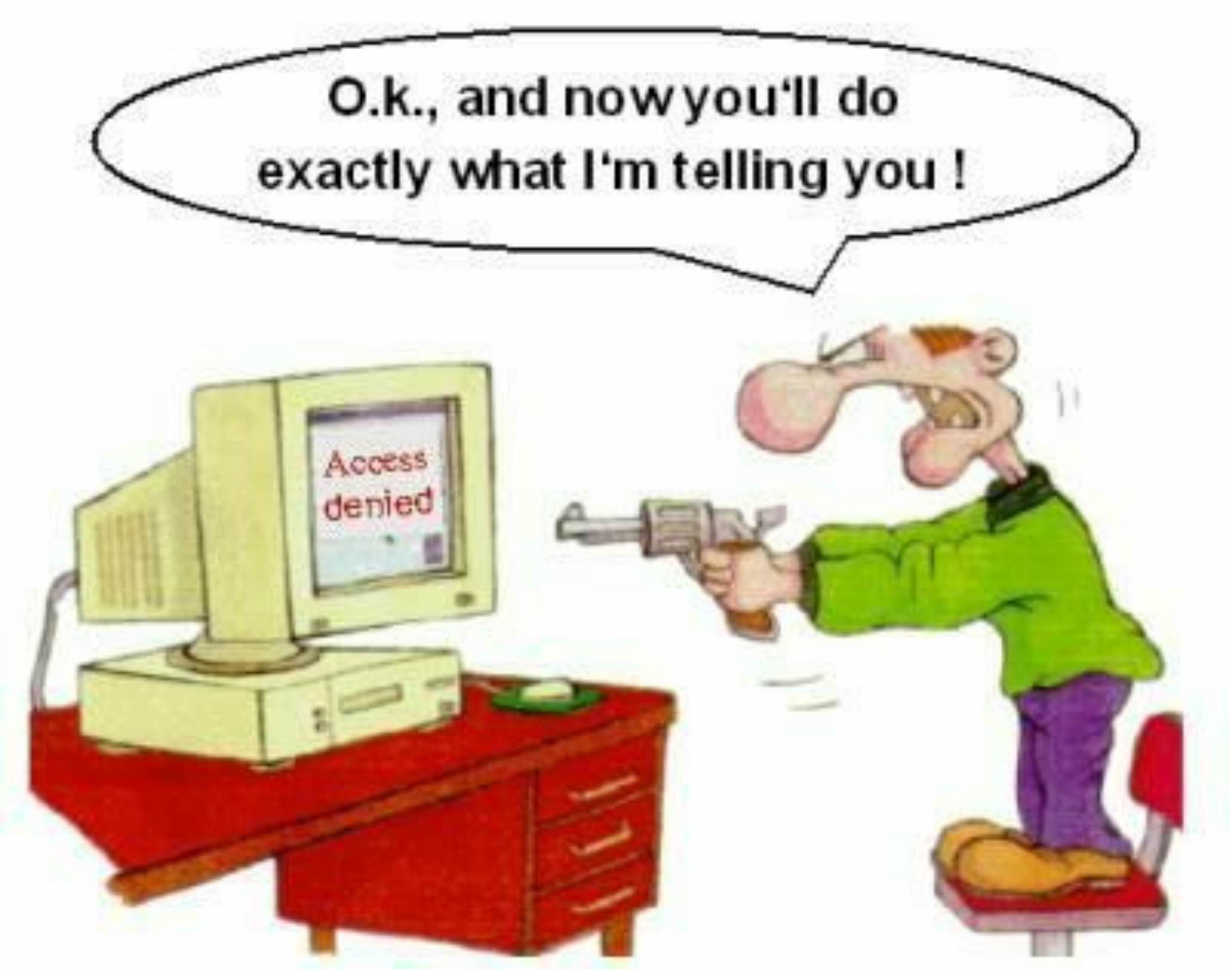
When a key is pressed on the keyboard, the key’s scan code appears in the keyboard memory map

<u>To check which key is currently pressed:</u>

- Probe the contents of the Keyboard chip
- In the Hack computer: probe the contents of RAM[24576]

Keyboard Input Demo

Things To Do


O.k., and now you'll do exactly what I'm telling you !
Access denied