การจัดองค์การคอมพิวเตอร์

W5.4 แฮกค์คอมพิวเตอร์

31110321 Computer Organization
สำหรับนักศึกษาชั้นปีที่ 3 สาขาวิชาวิศวกรรมคอมพิวเตอร์

ทรงฤทธิ์ กิติศรีวรพันธุ์

songrit@npu.ac.th

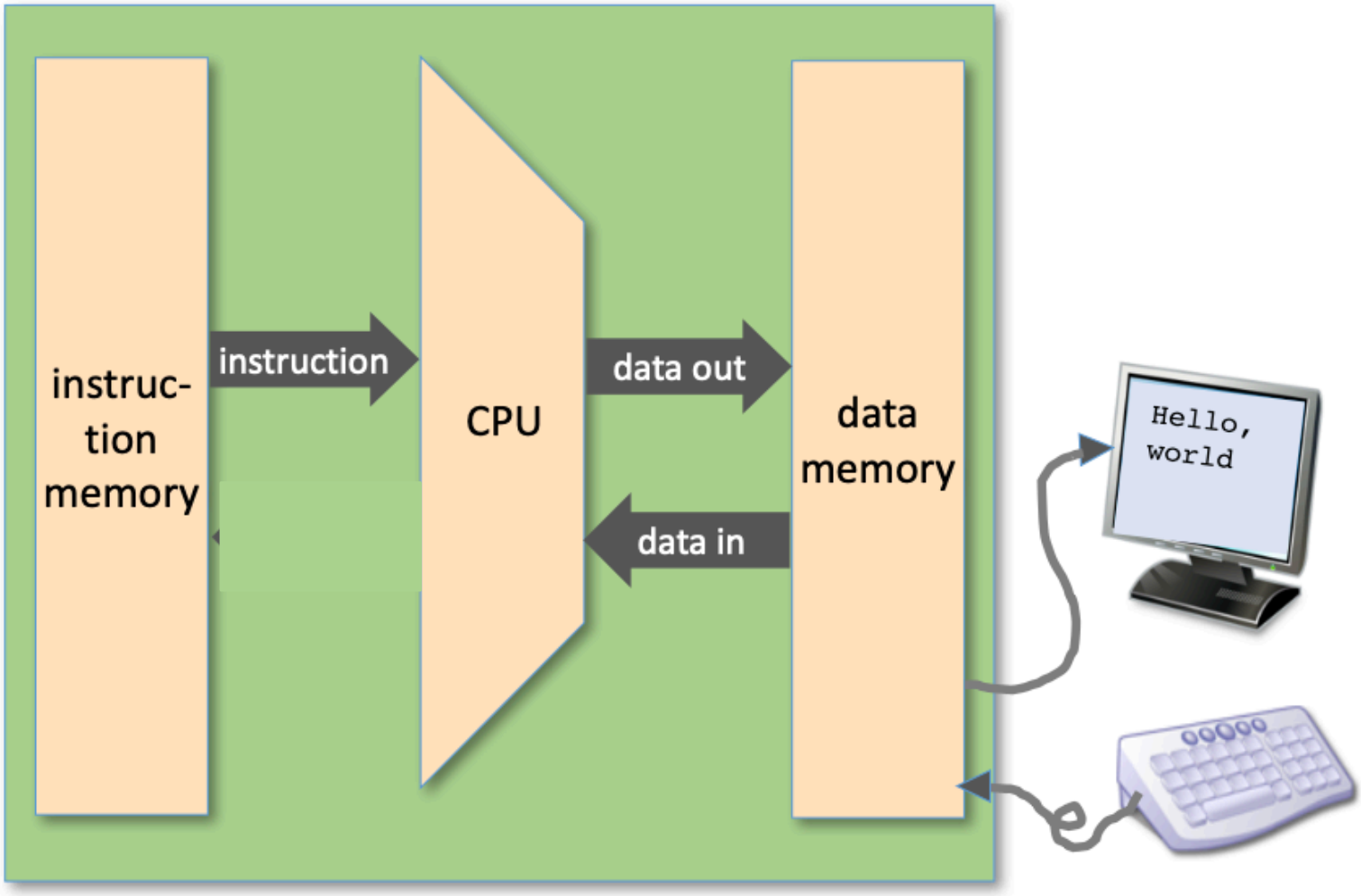
สาขาวิชาวิศวกรรมคอมพิวเตอร์
มหาวิทยาลัยนครพนม

Revised 2020-09-21

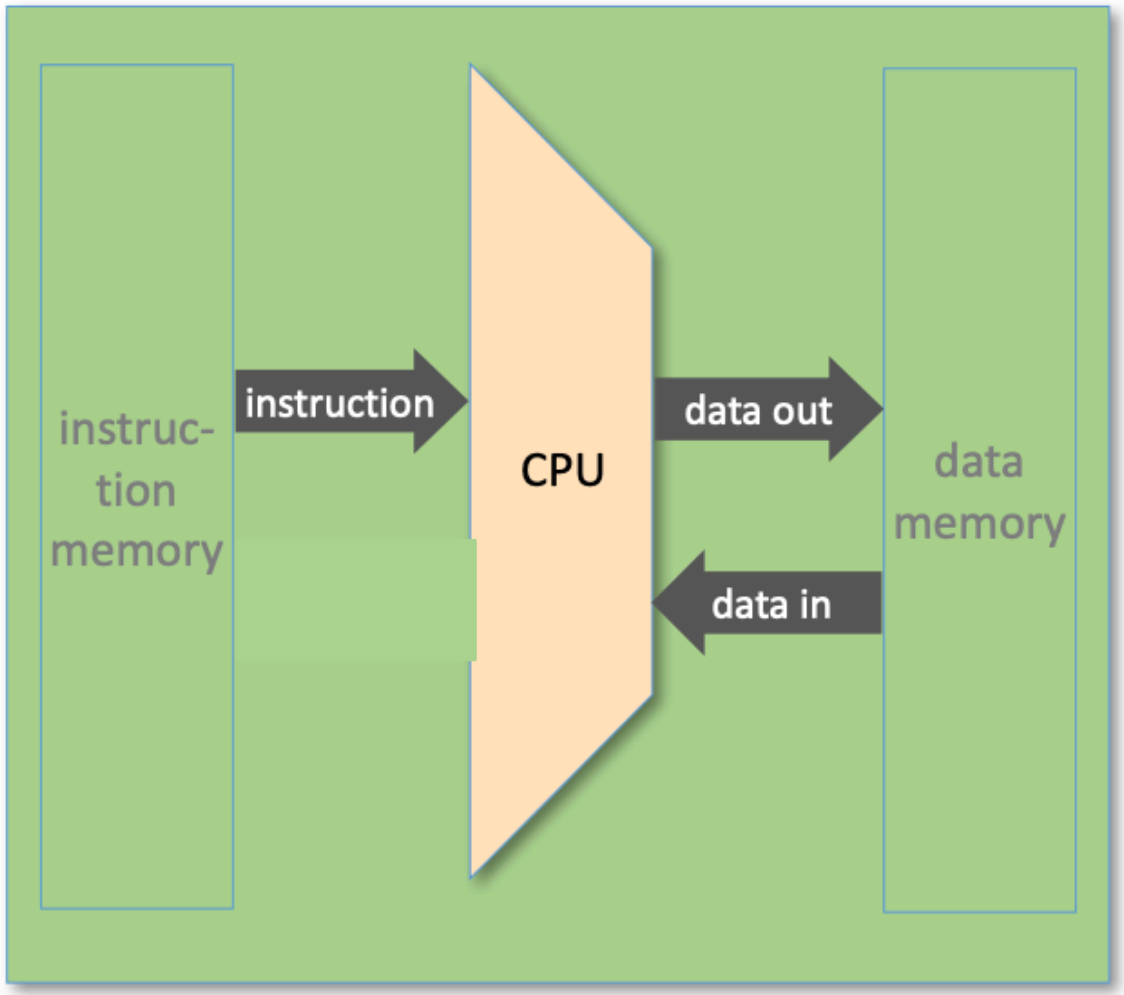
Lecture plan

- 5.1 สถาปัตยกรรมฟอนนอยมันน์
- 5.2 Fetch-Execute Cycle
- 5.3 ซีพียูแฮกค์
- **5.4 แฮกค์คอมพิวเตอร์**
- 5.5 ภาพรวมโปรเจ็ค 5

Hack Computer

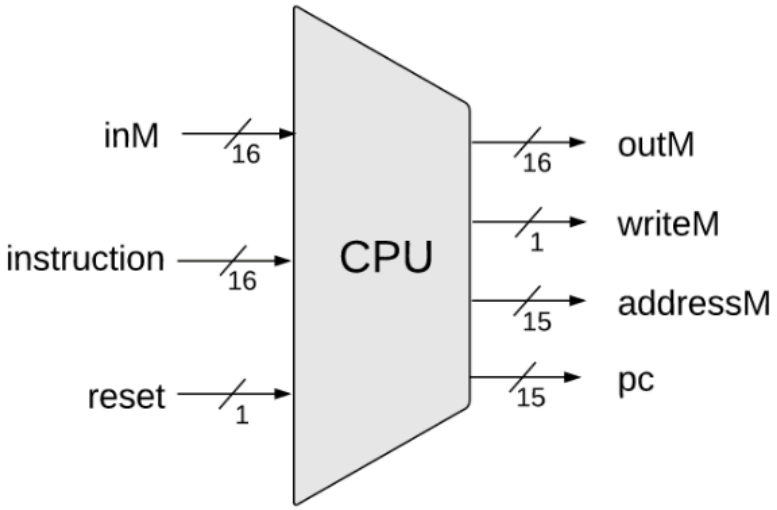


Hack CPU


instruc-
tion
memory
instruction
CPU
data out
data in
data
memory

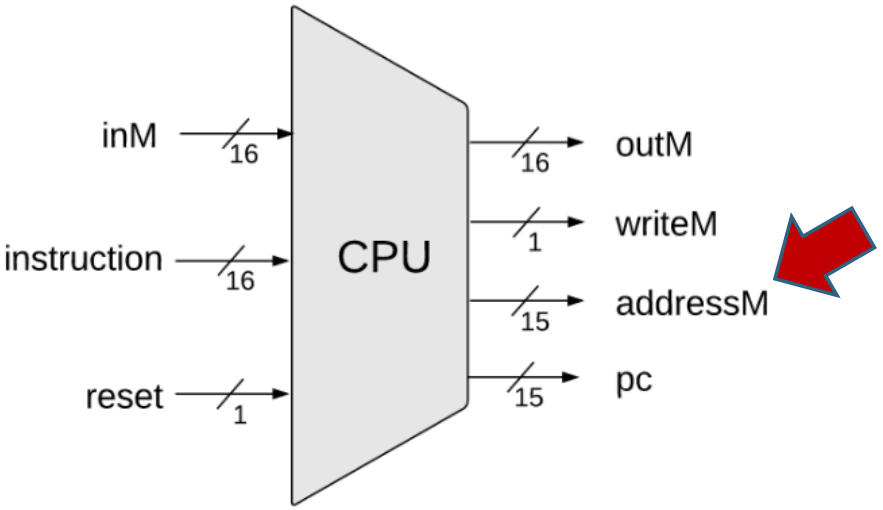
Hack CPU Operation

```
D=D-A
@17
M=M+1
```



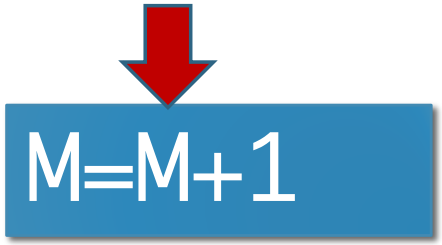
- เมื่อคำสั่งมี D และ A; CPU จะติดต่อ DRegister และ ARegister ที่อยู่ภายใน CPU

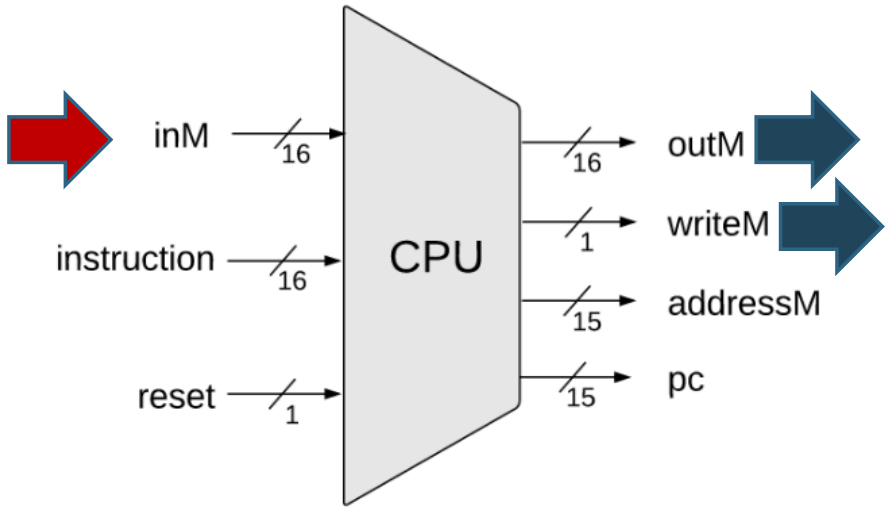
Hack CPU Operation



- @x (A-instruction) CPU จะนำ x เก็บลง A
- และค่านี้คือ addressM

Hack CPU Operation

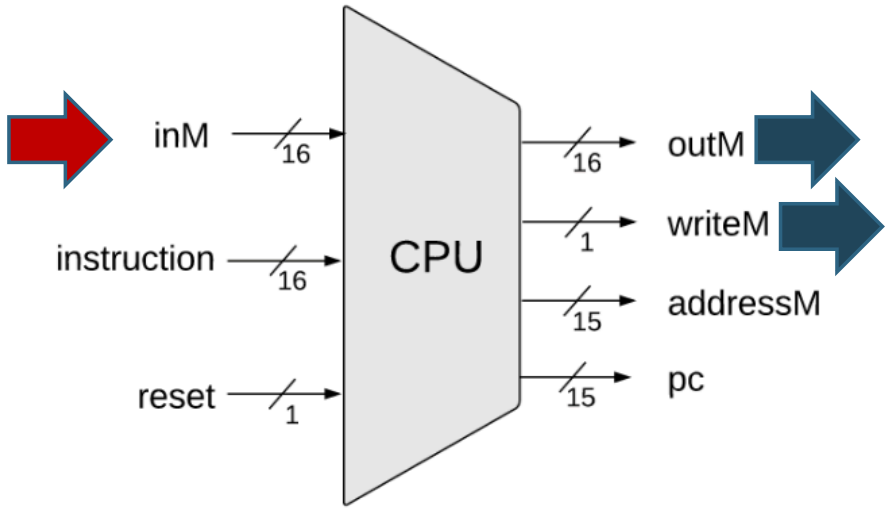




- ถ้า**ขวามือ**(RHS) มีนีโมนิค M ซีพียูค่าอ่าน inM
- **ซ้ายมือ**(LHS) มีนีโนนิค M
  - ส่งออก outM
  - เซต writeM เป็น 1

Hack CPU Operation : Jump

```
@100
D=D-1; JEQ
```

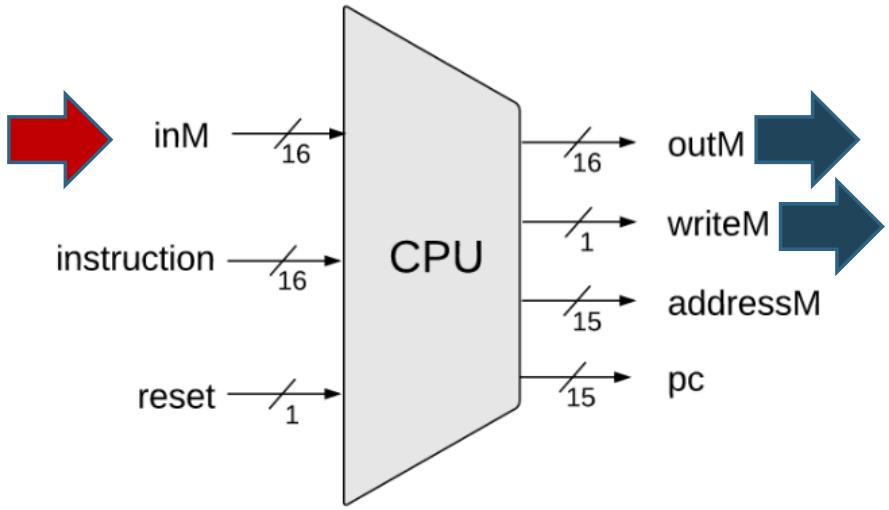


- ARegister เก็บ 100

If (reset==0)
เป็นภาวะรันโปรแกรมปกติ CPU พิจารณาว่าจะ Jump หรือไม่
If Jump: เซต PC=A
Else: PC++

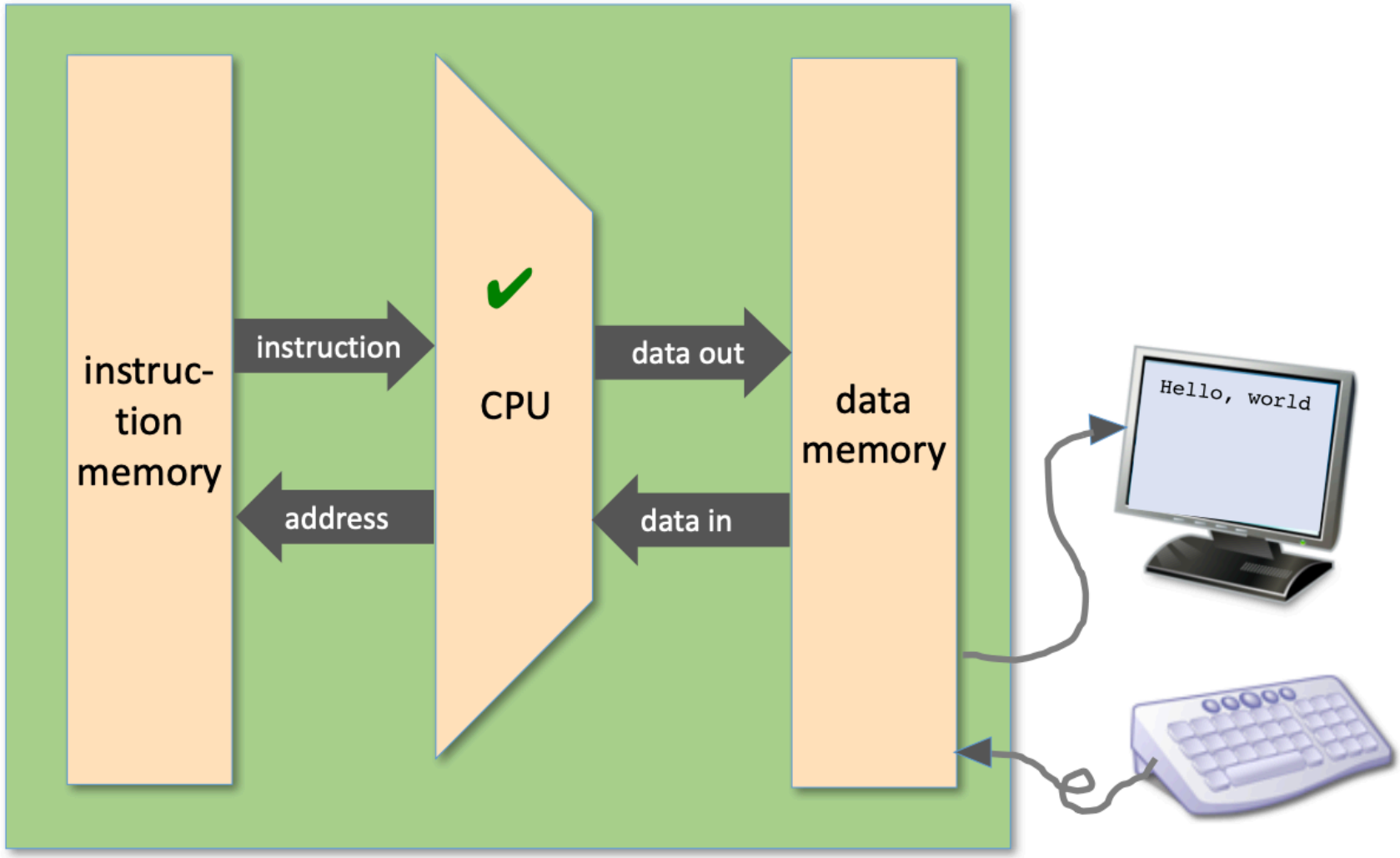
Hack CPU Operation : Jump

```
@100
D=D-1; JEQ
```

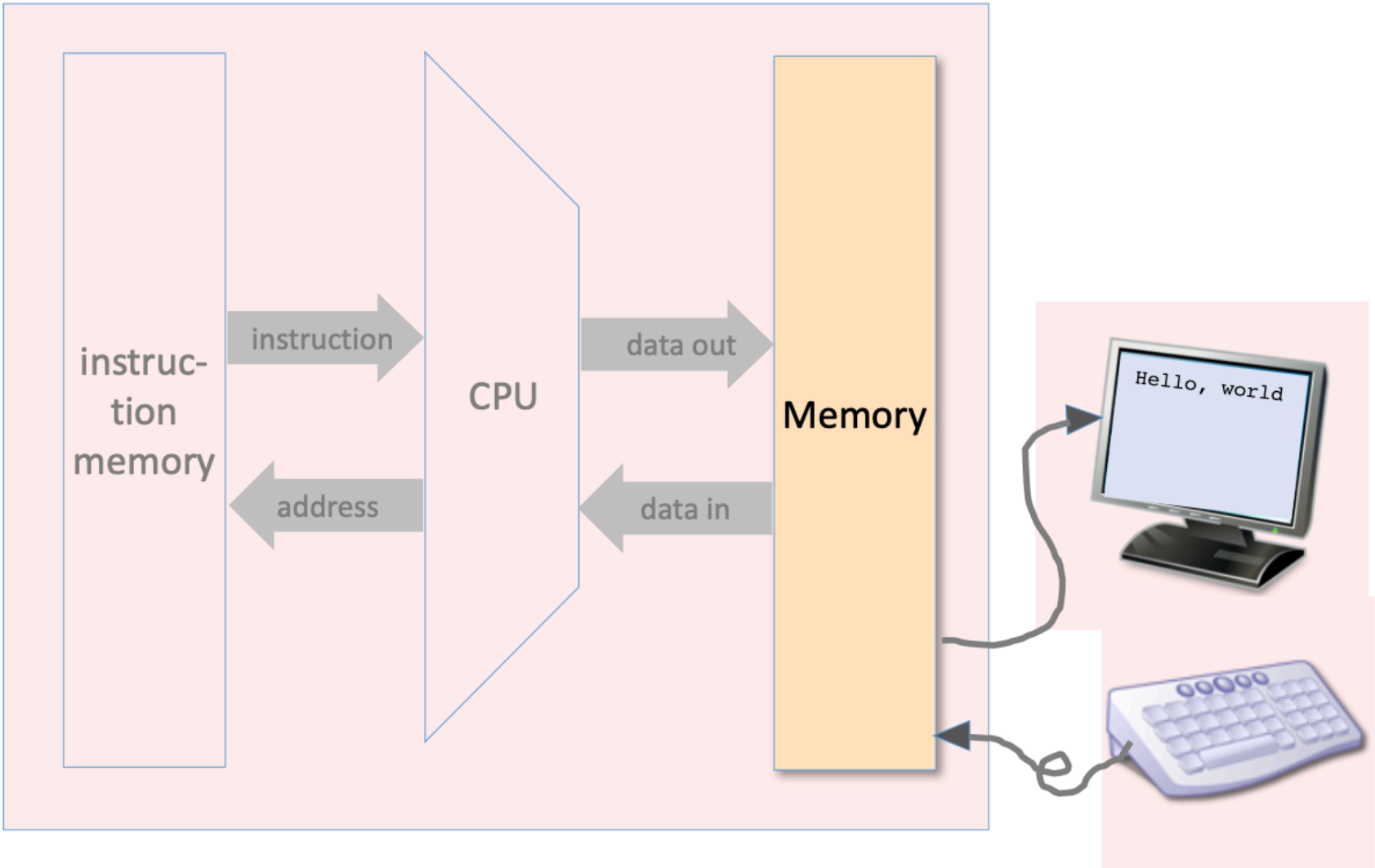


If (reset==1)
หมายถึง user กดปุ่ม reset ต้องการเริ่มโปรแกรมใหม่
PC=0

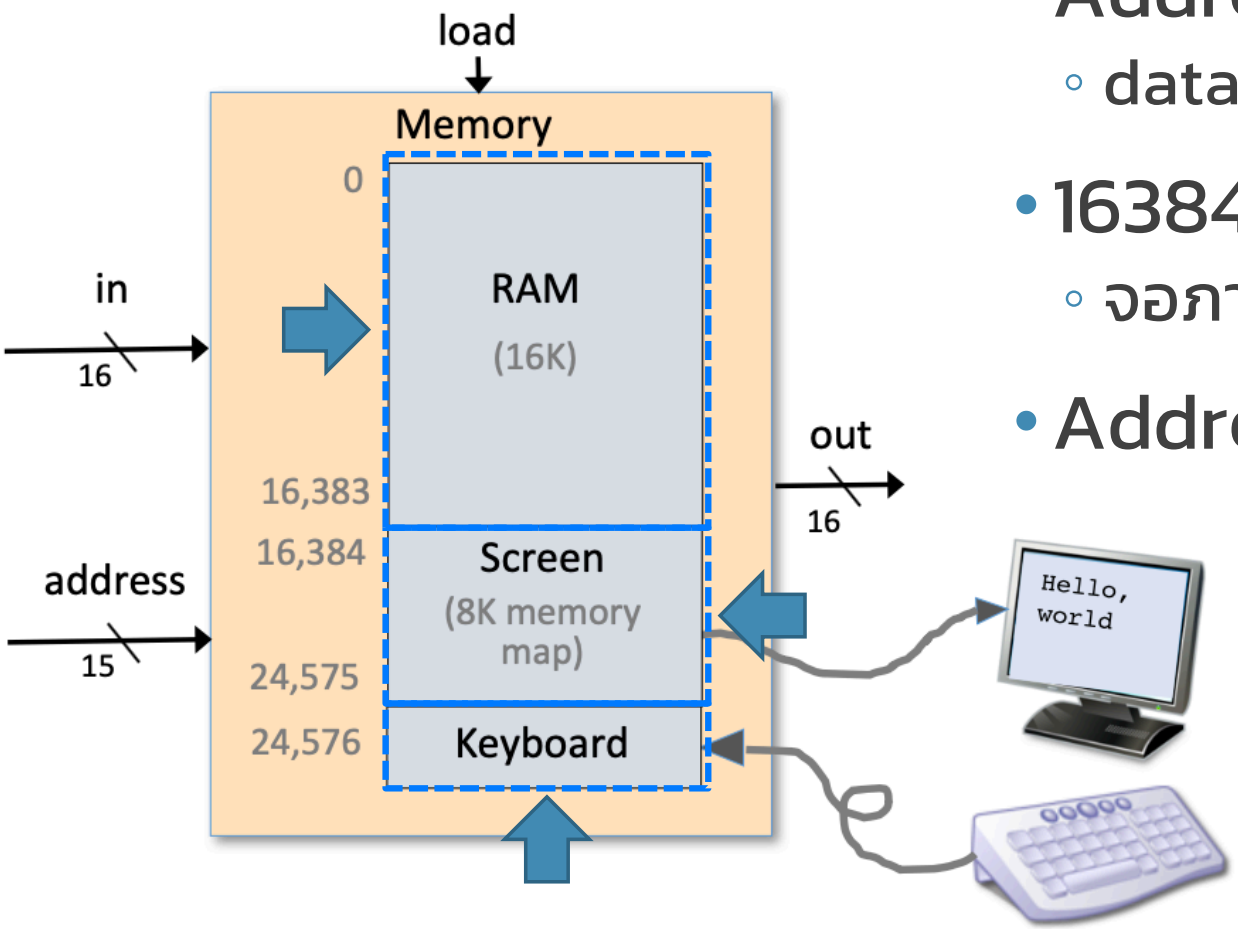
แฮกค์คอมพิวเตอร์



Memory


instruc-
tion
memory
instruction
address
CPU
data out
data in
Memory
Hello, world

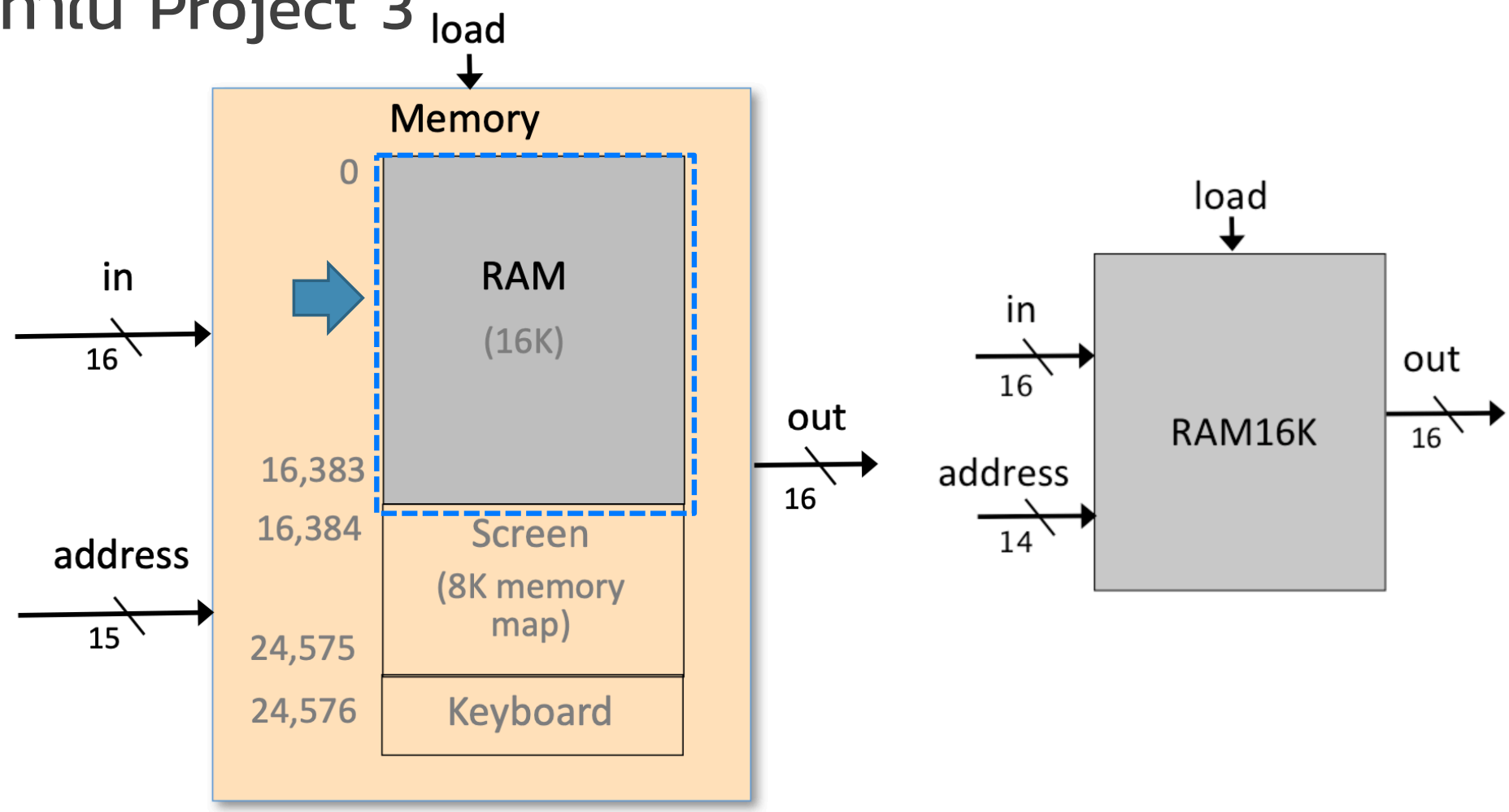
Memory : คอนเซ็ป



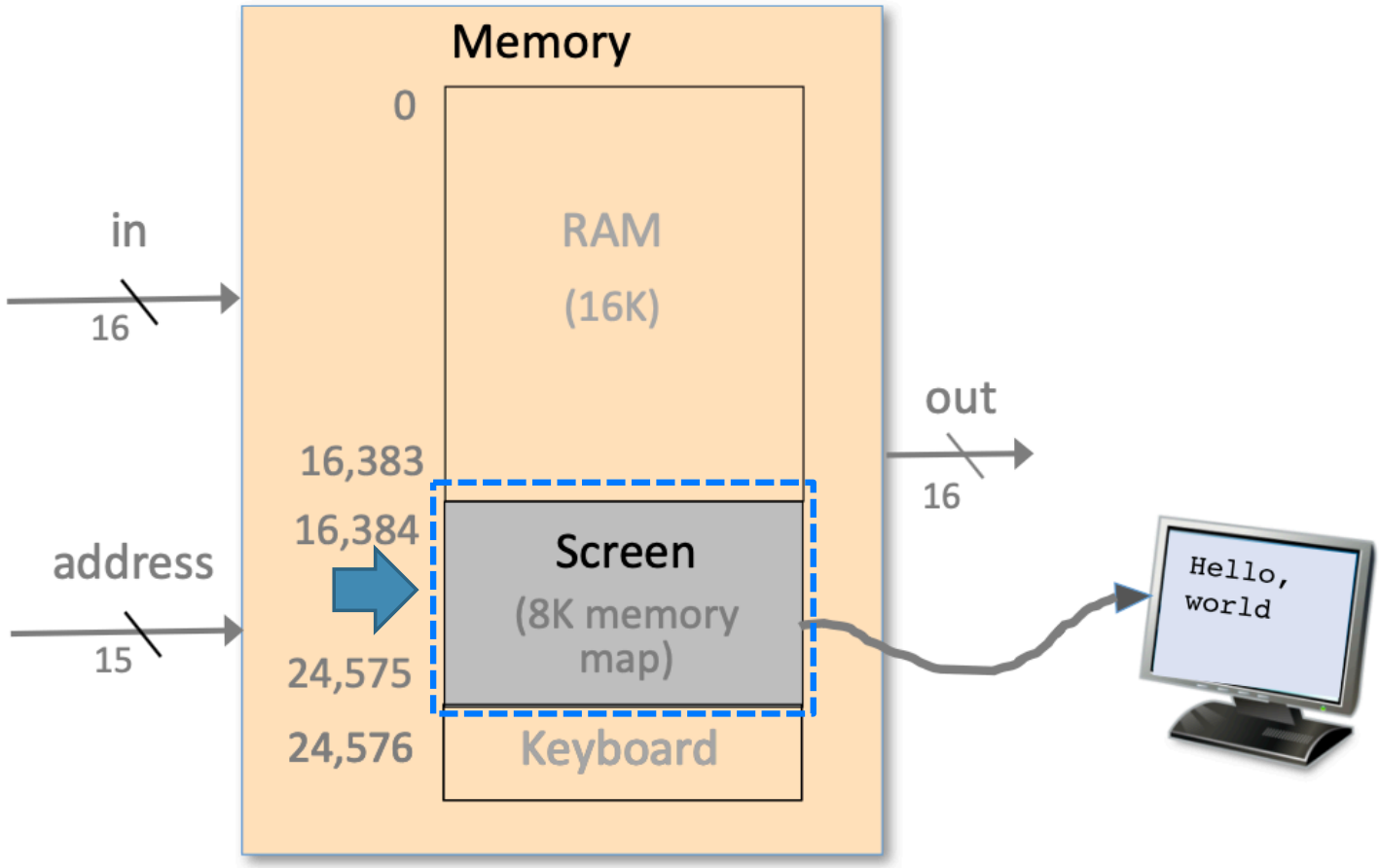
- Address 0 ถึง 16383
  - data memory
- 16384 ถึง 24575
  - จอภาพ
- Address 24576 →คีย์บอร์ด

Memory : implement

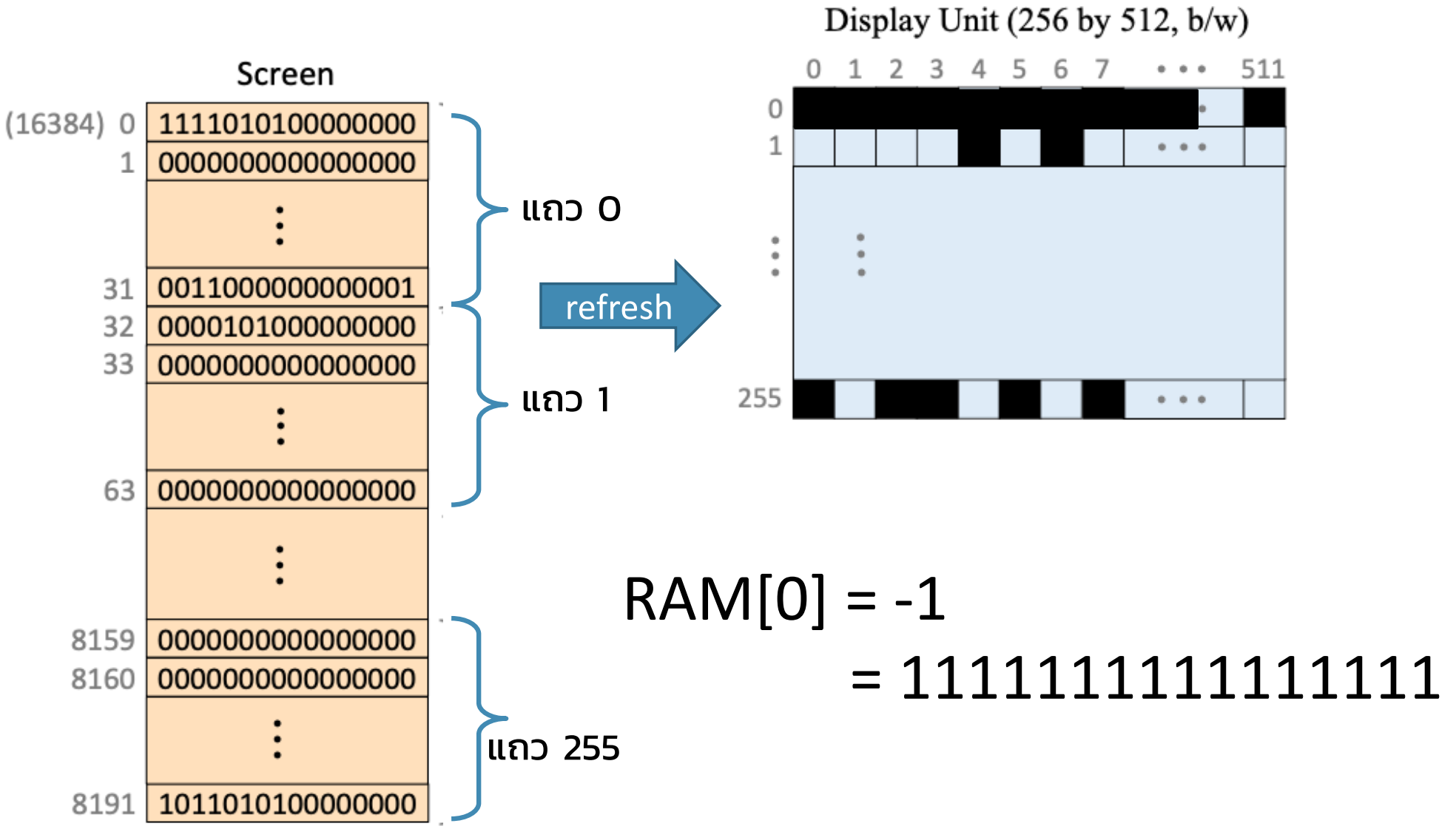
- ทำใน Project 3



Screen


Memory
0
RAM
(16K)
16,383
16,384
Screen
(8K memory
map)
24,575
24,576
Keyboard
in
16
address
15
out
16
Hello,
world

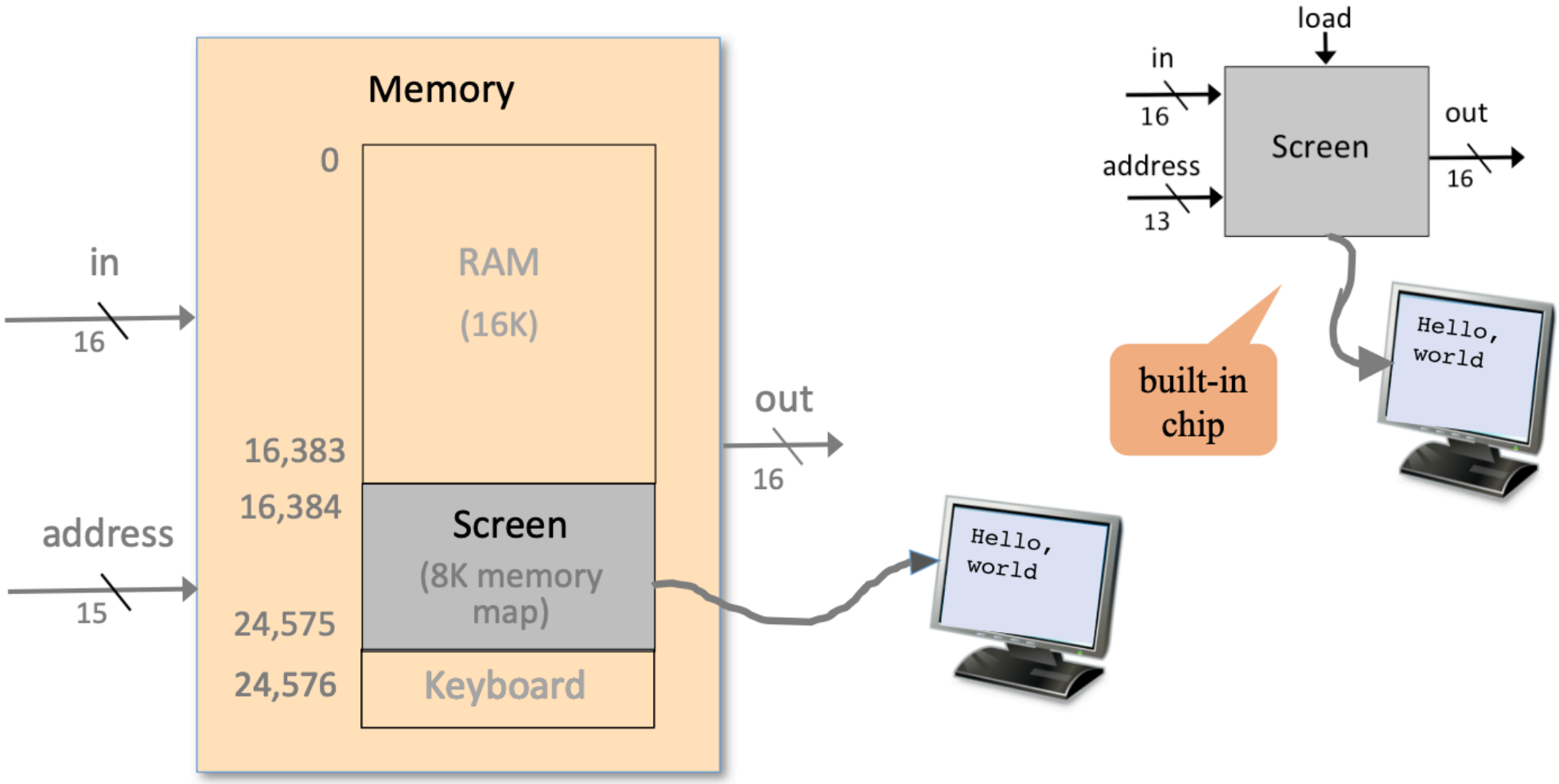
Memory mapped output



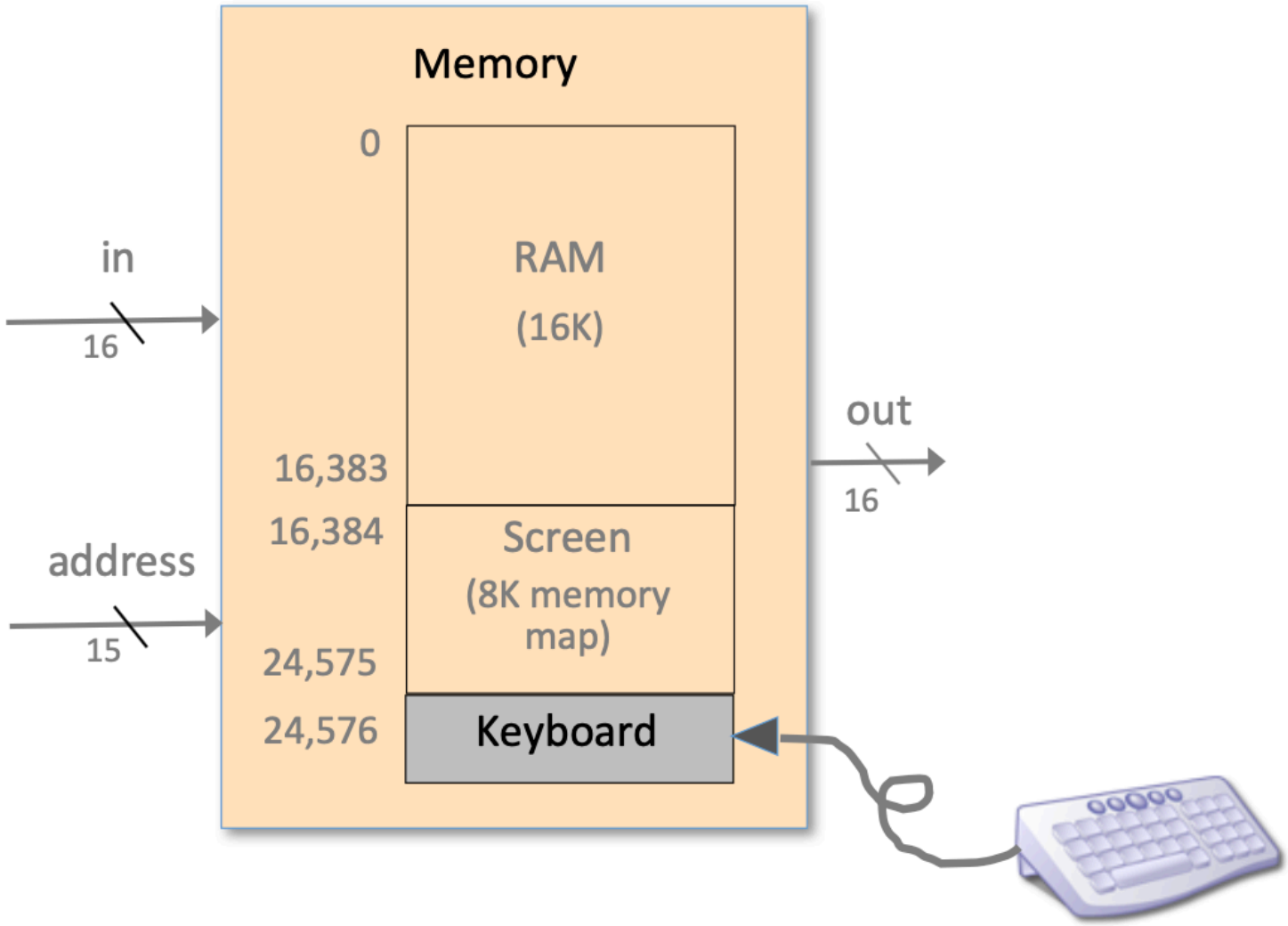
RAM[0] = -1

= 1111111111111111

Screen


Memory
0
RAM
(16K)
16,383
16,384
Screen
(8K memory map)
24,575
24,576
Keyboard
in
16
address
15
out
16
load
in
16
address
13
Screen
out
16
built-in chip
Hello, world
Hello, world

Keyboard


Memory
0
RAM
(16K)
16,383
16,384
Screen
(8K memory map)
24,575
24,576
Keyboard
in
16
address
15
out
16

Keyboard memory map

Keyboard

0000000001001011

Scan-code of 'k' = 75

Keyboard 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

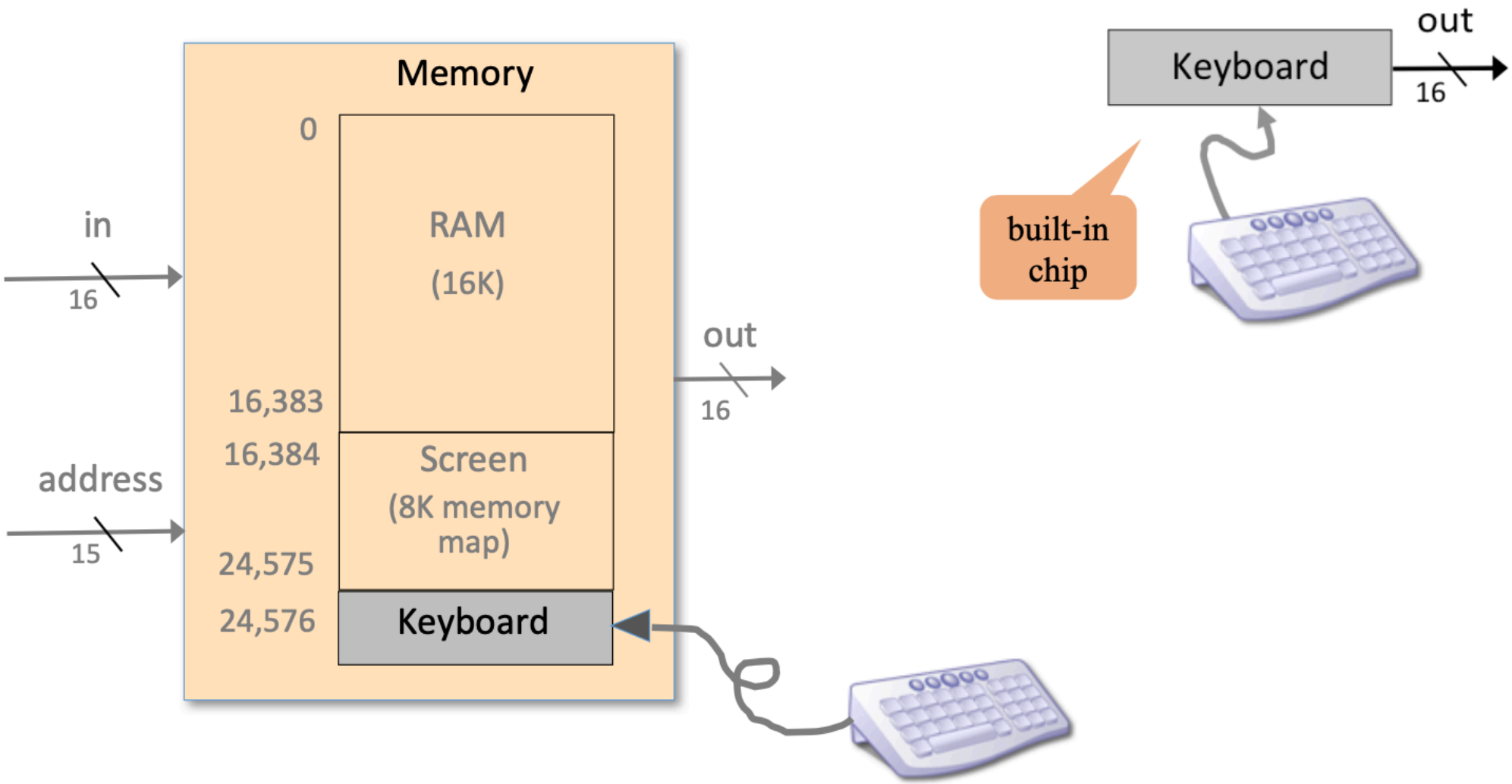
[	91
/	92
]	93
^	94
_	95
`	96

key	code
a	97
b	98
c	99
…	…
z	122

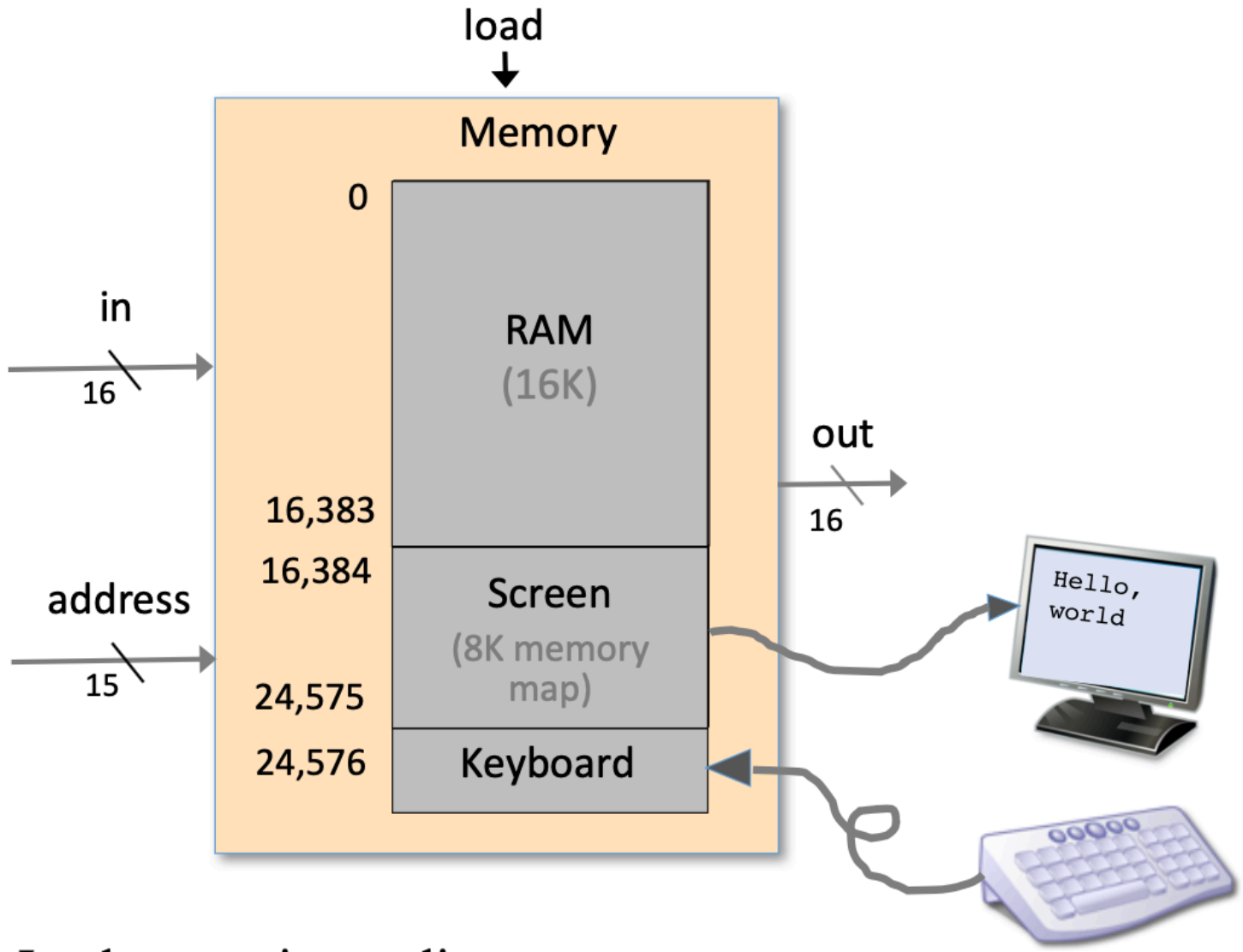
{	123
|	124
}	125
~	126

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

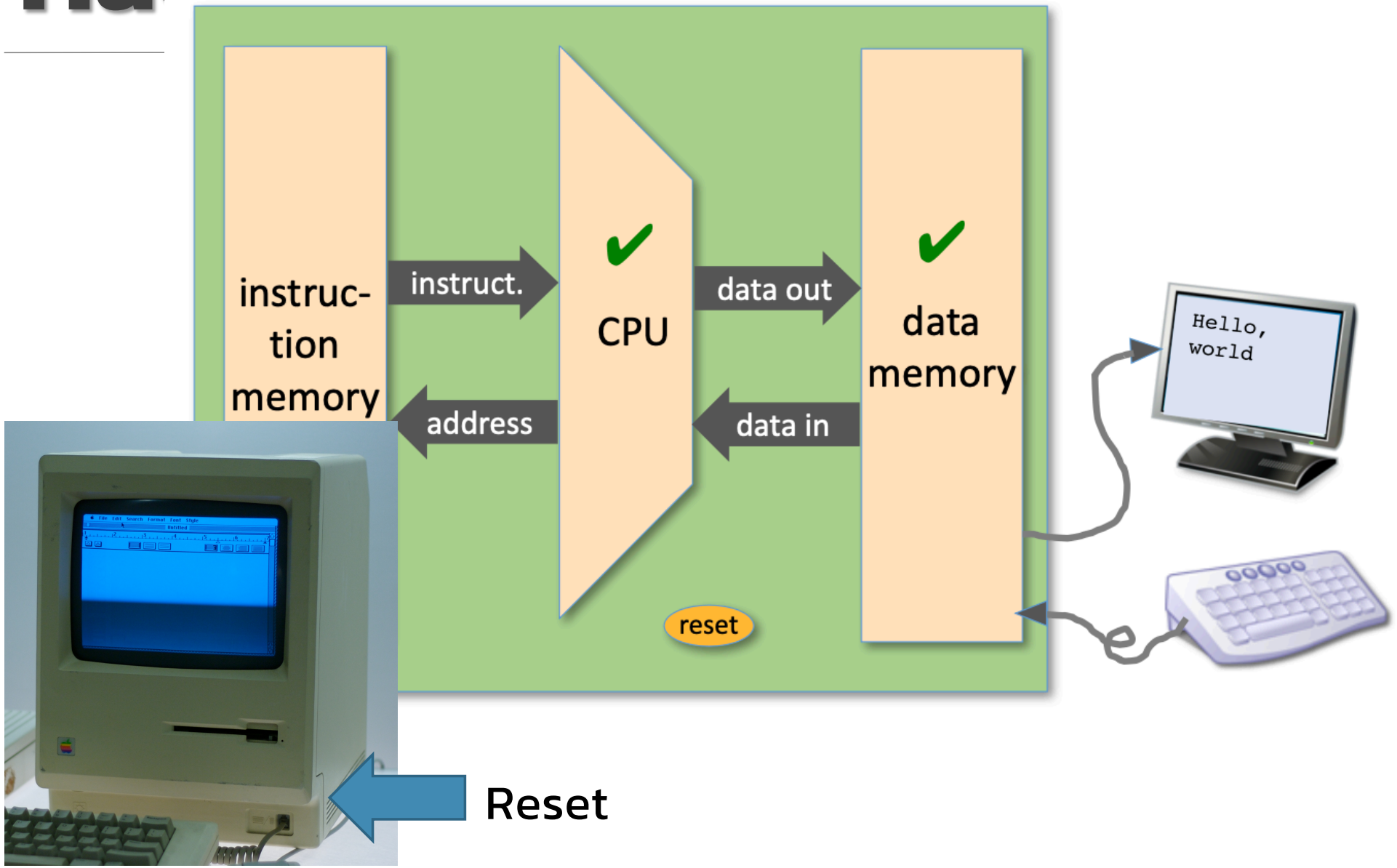
Keyboard


Memory
0
RAM
(16K)
16,383
16,384
Screen
(8K memory
map)
24,575
24,576
Keyboard
in
16
address
15
out
16
Keyboard
out
16
built-in
chip

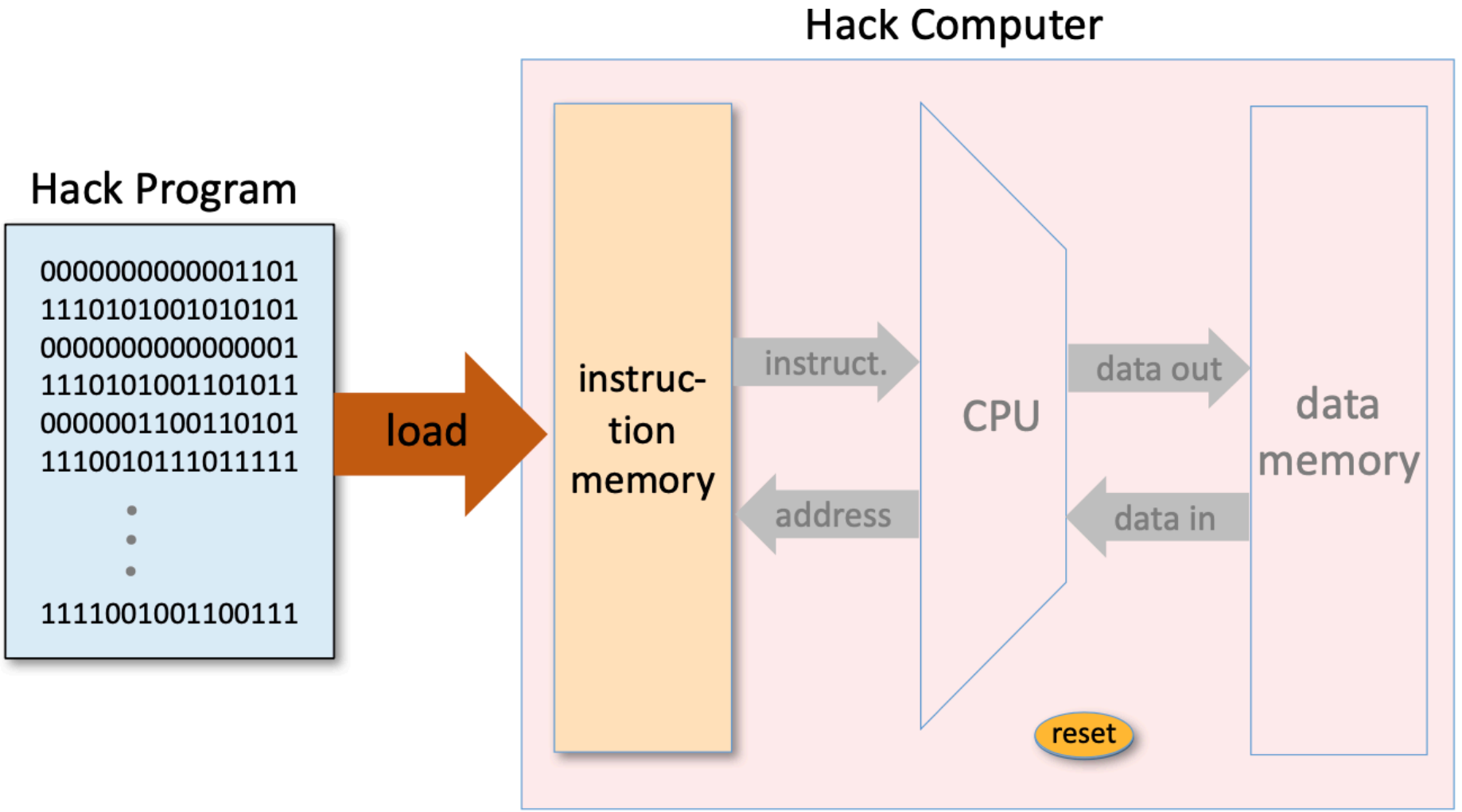
Memory implementation



Hack Computer

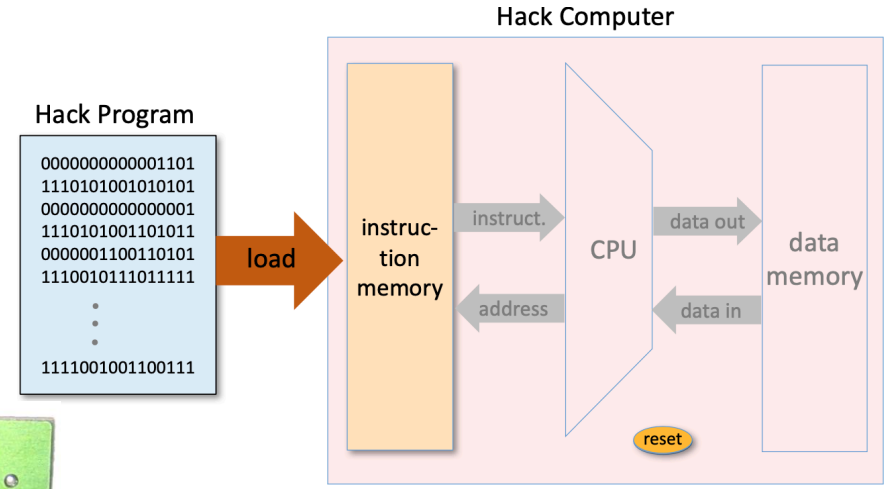


Instruction memory



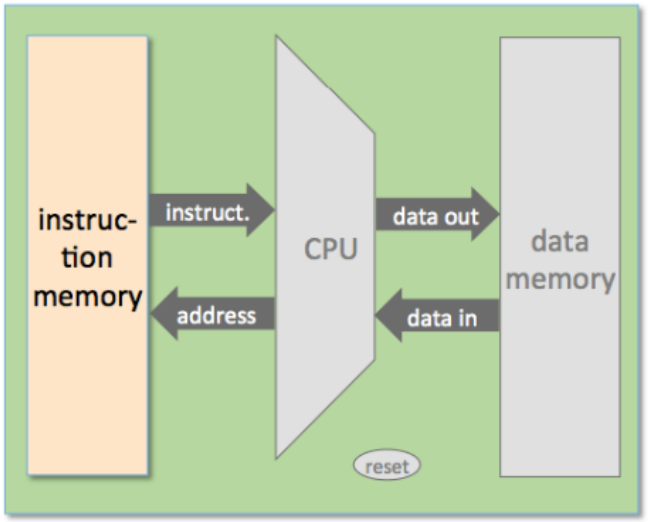
Instruction memory

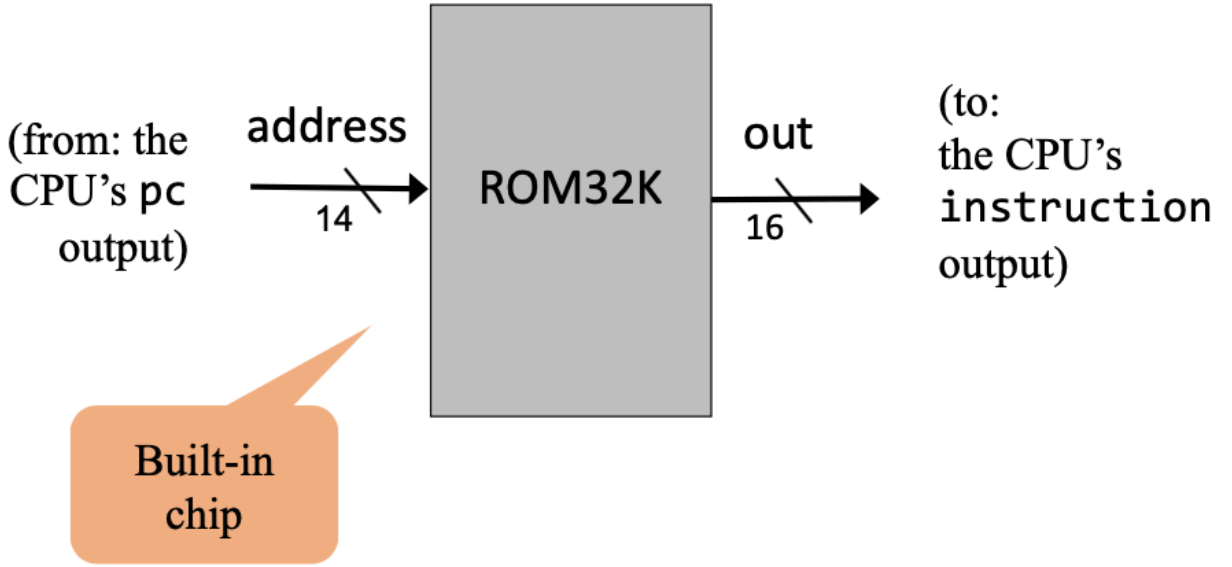
- Hardware implement
  - ใส่ instruction ใน ROM chip



- Hack CPU simulation
  - instruction เก็บเป็น text file

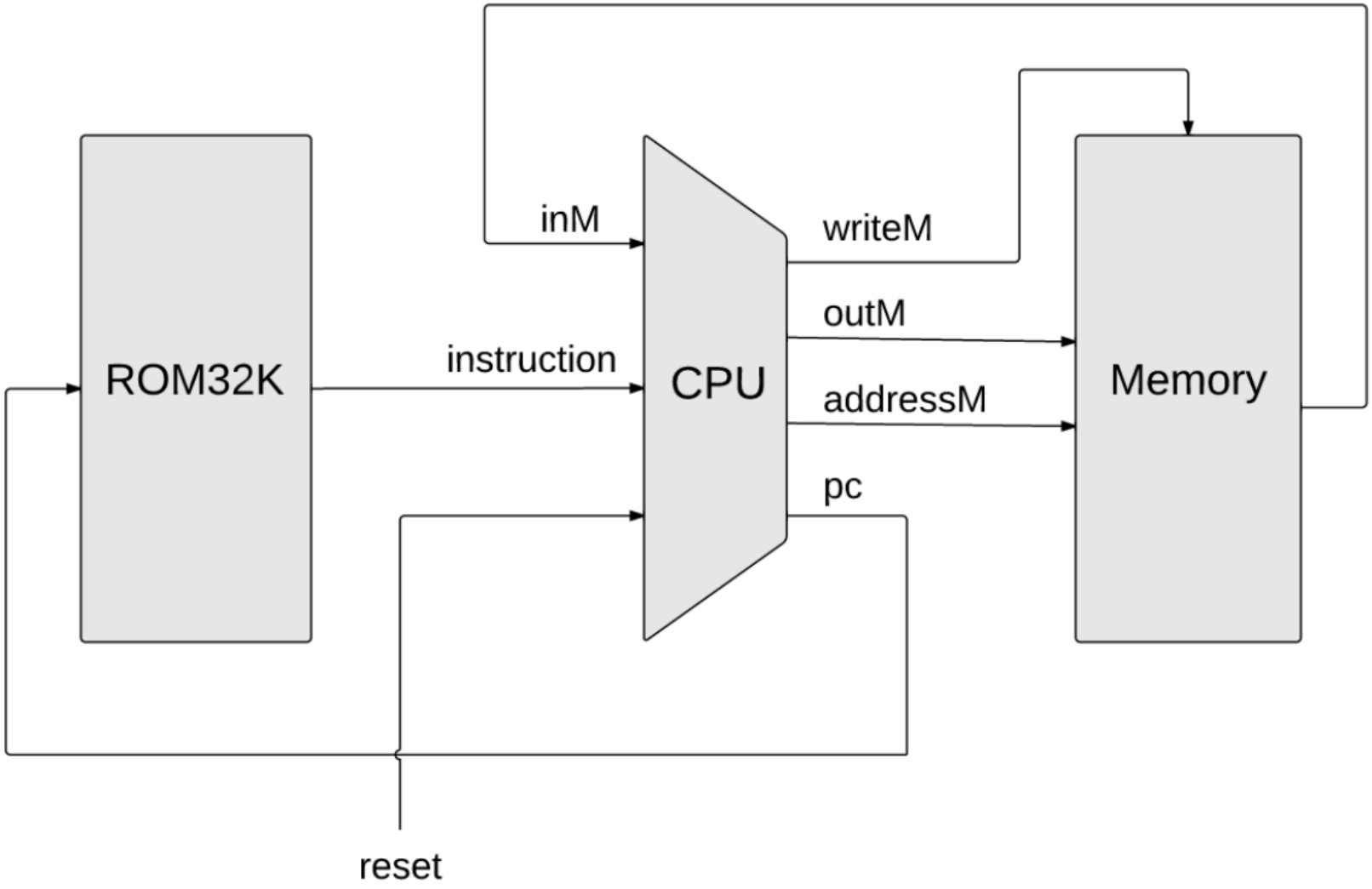
Instruction memory

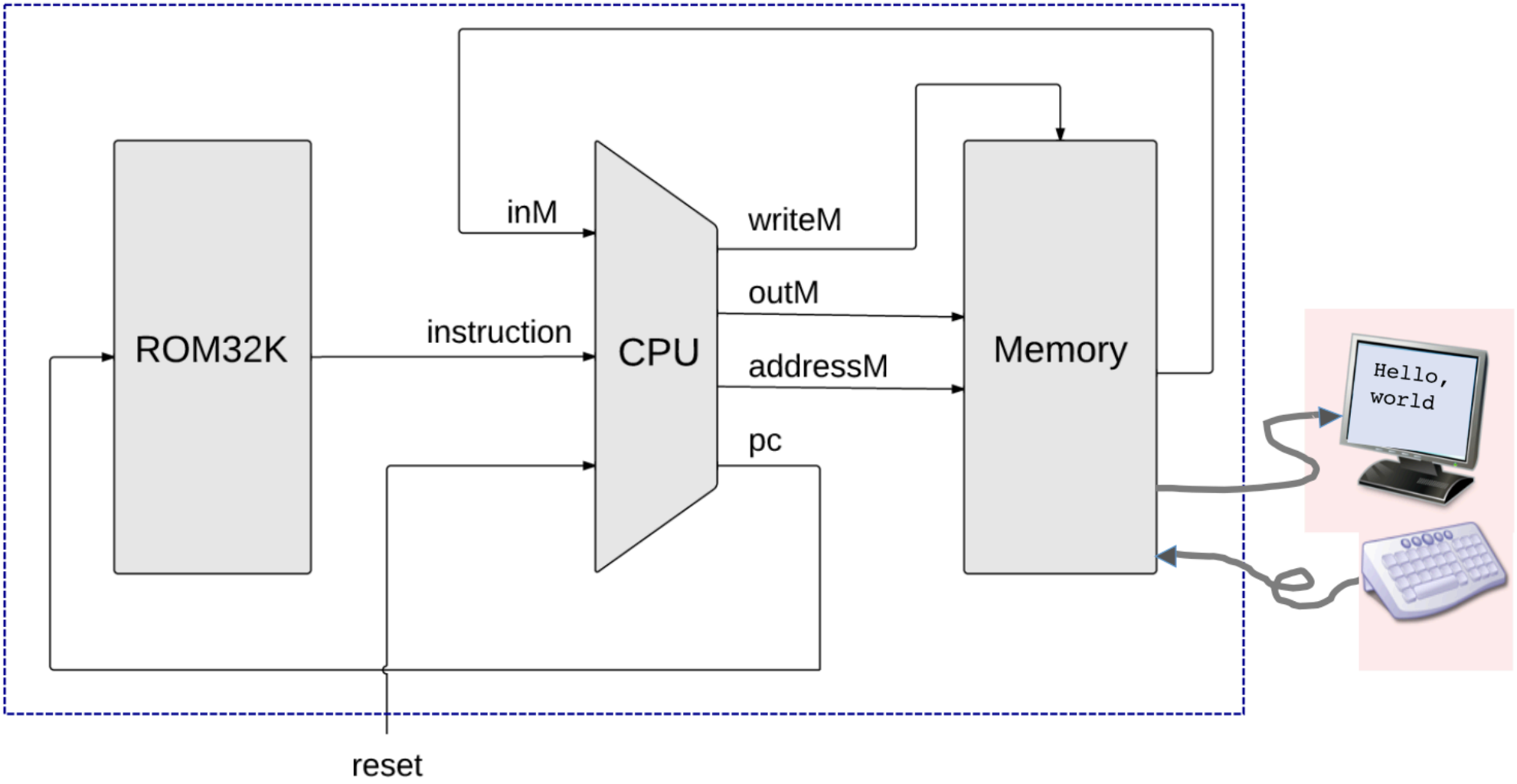




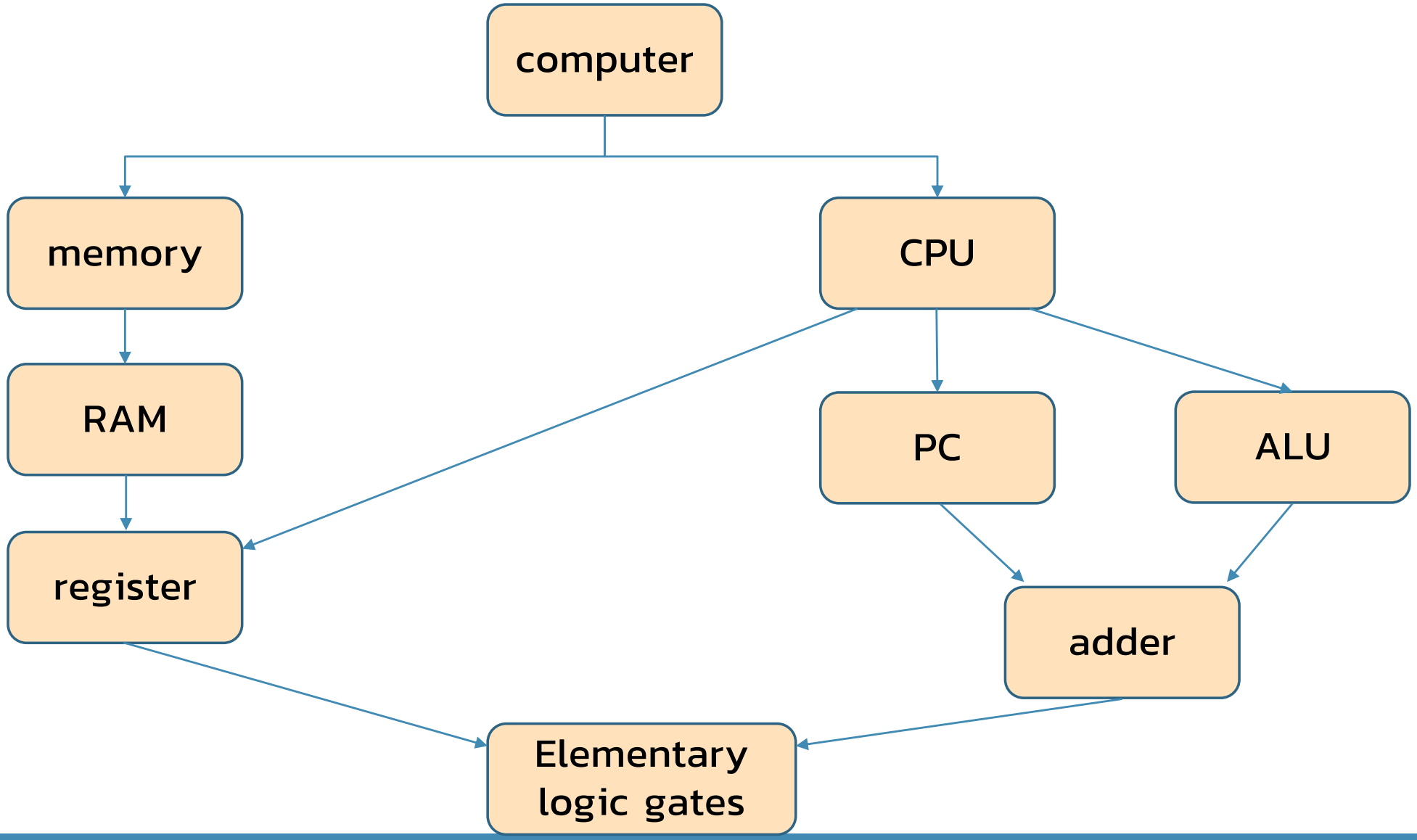
- Hack instruction Memory
  - Build-in chip ชื่อ ROM32K
  - ROM32K: เป็น read-only , 16-bit, 32K RAM chip

Hack Computer implementation

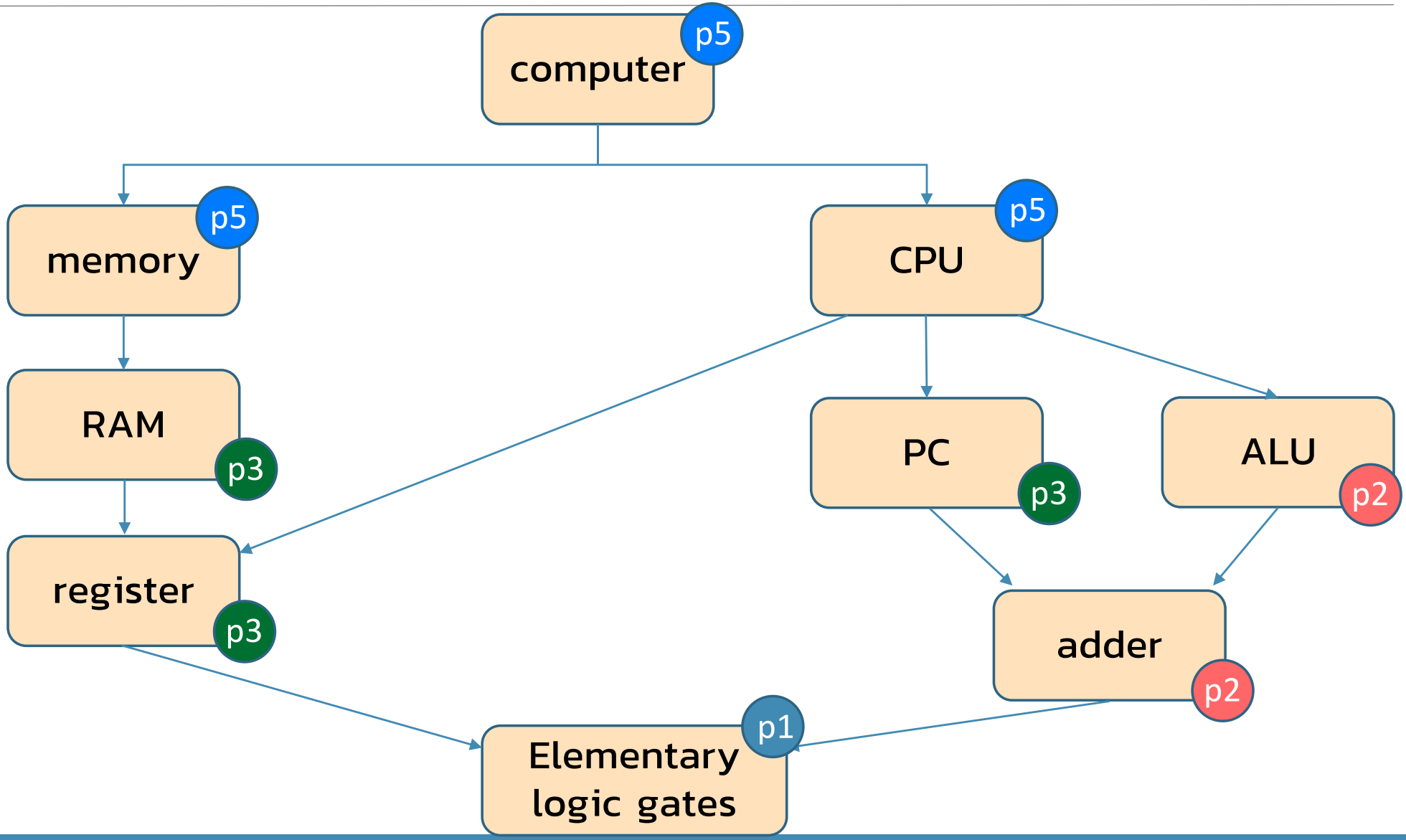


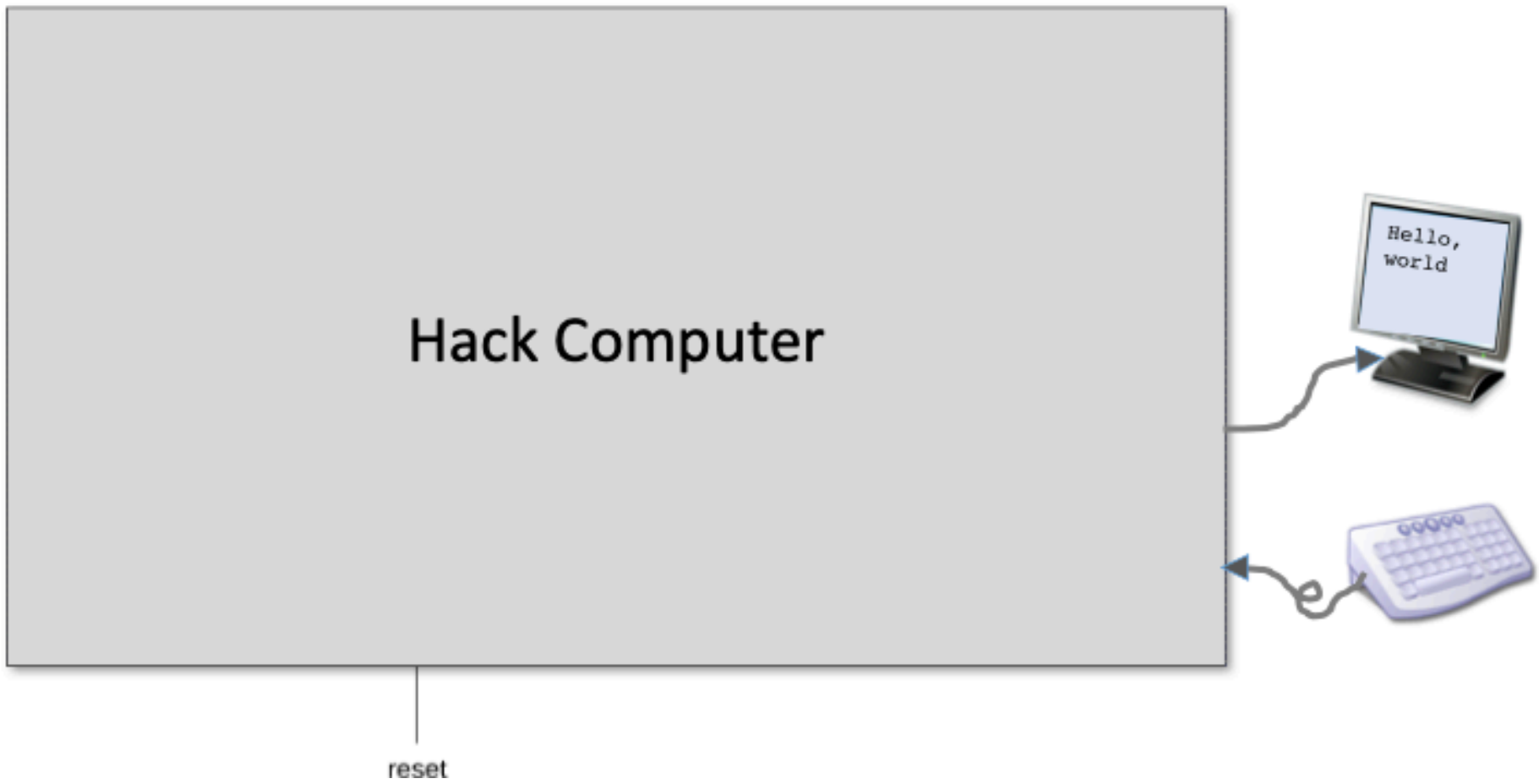

ROM32K
instruction
CPU
inM
writeM
outM
addressM
pc
Memory
reset
Hello,
world

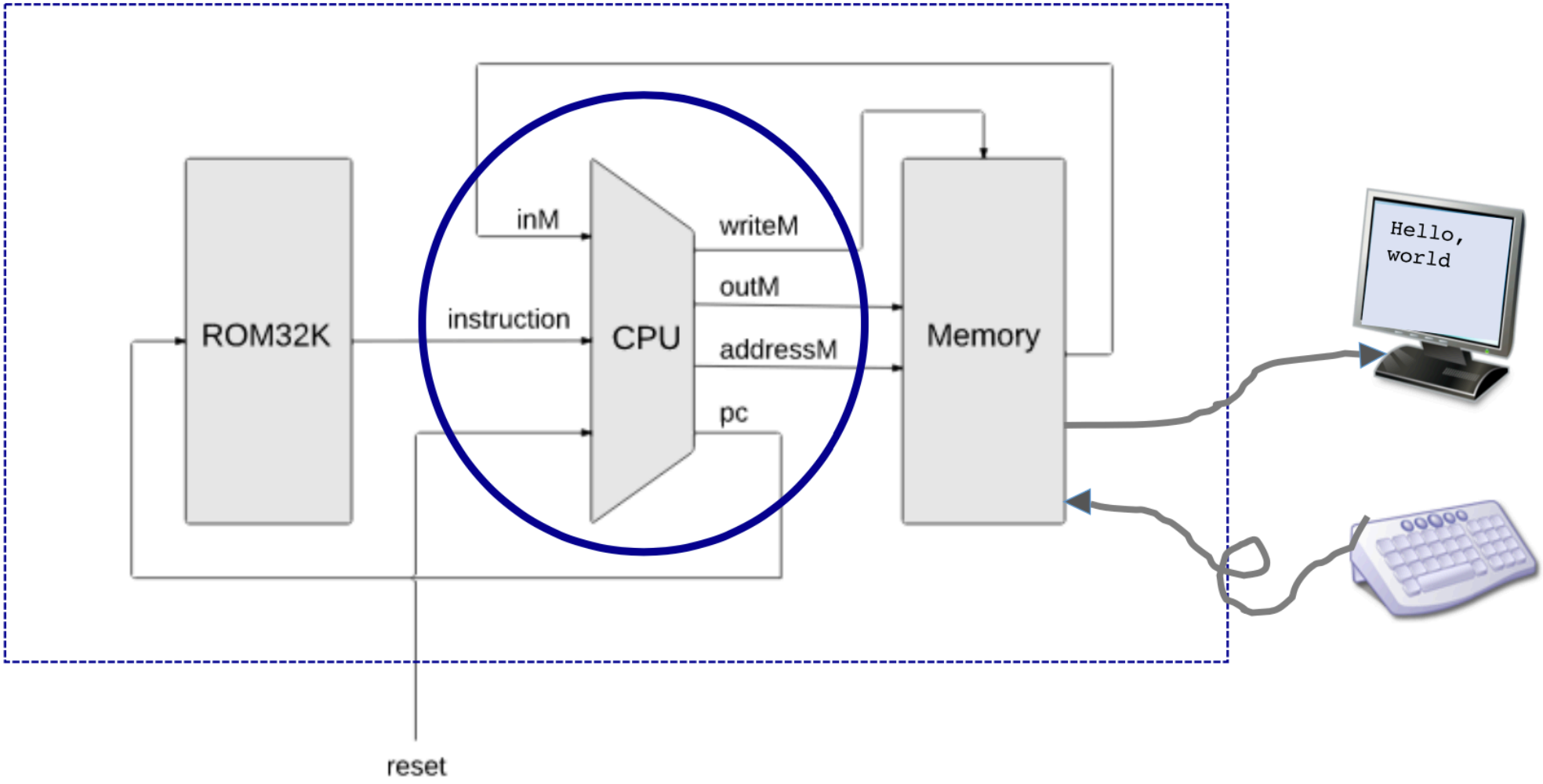
Hardware organization

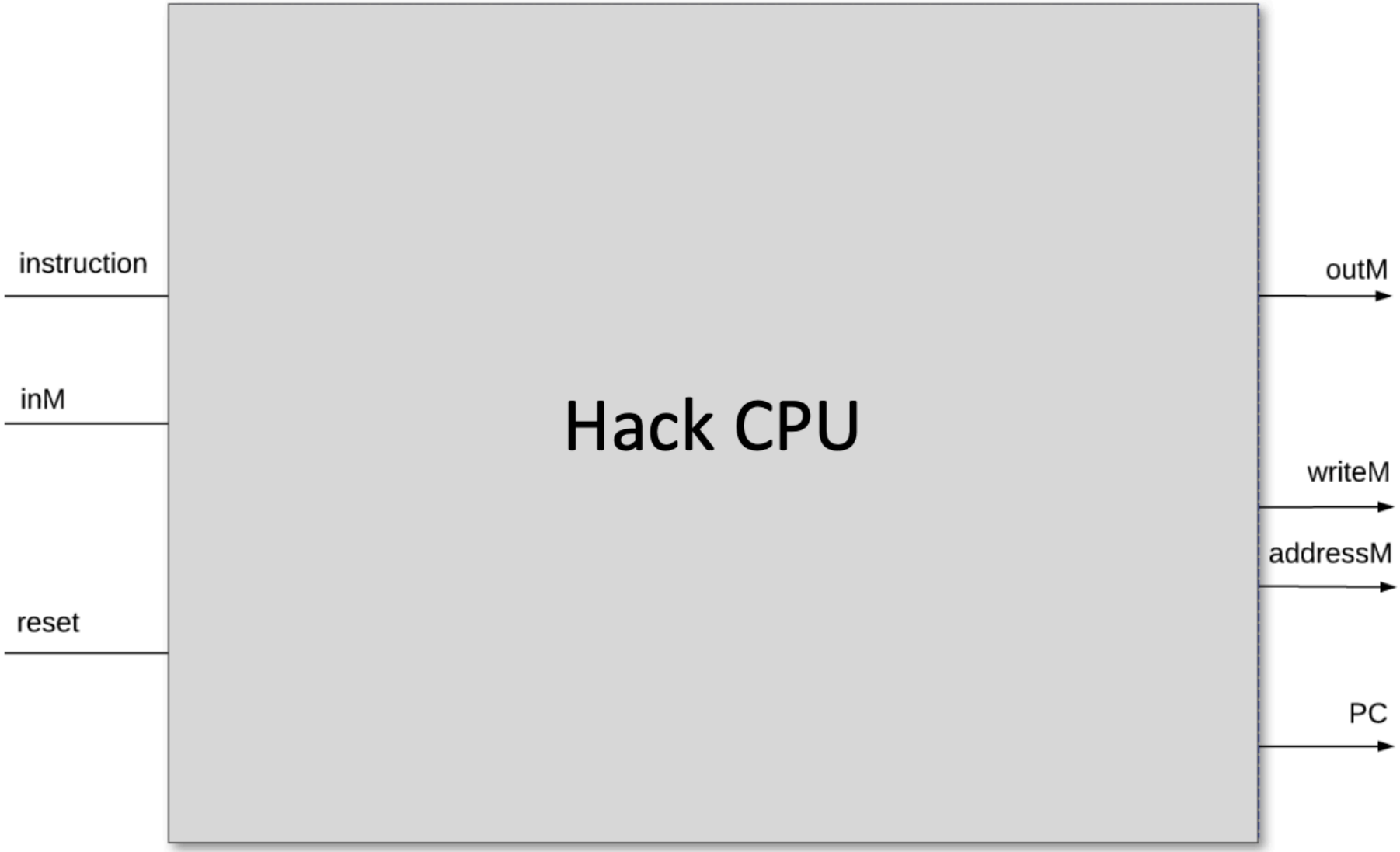


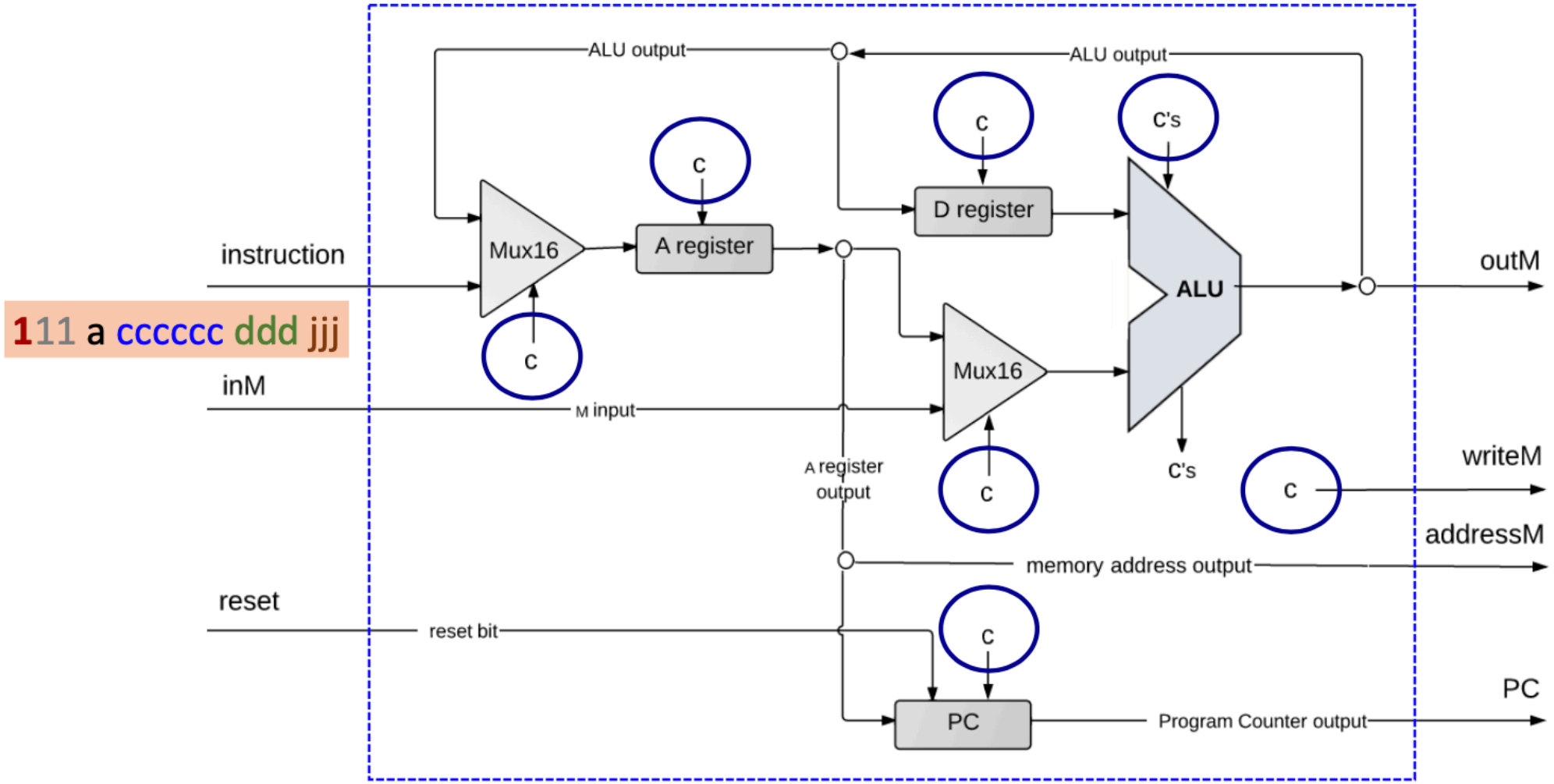
Hardware organization



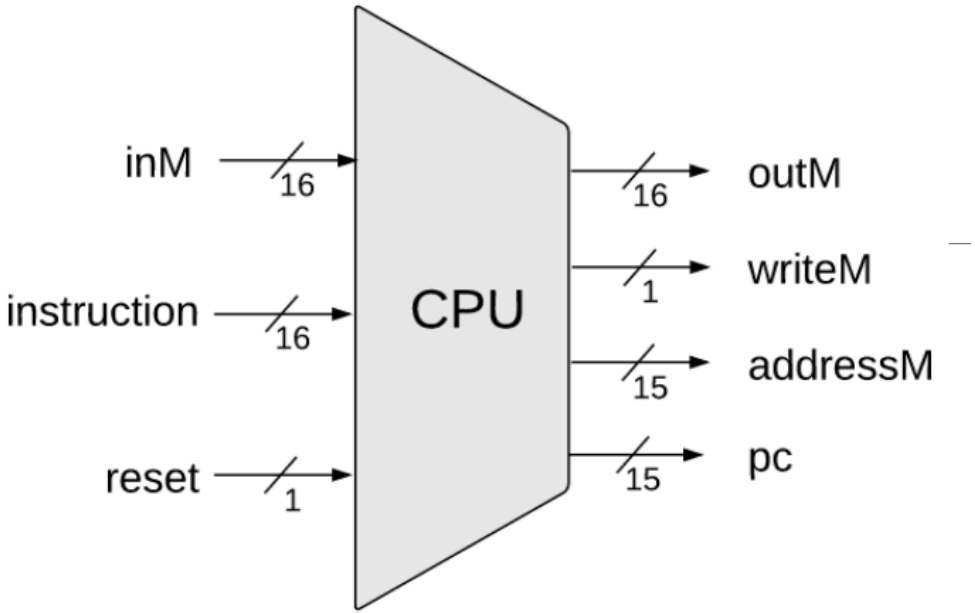

Hack Computer
Hello,
world
reset

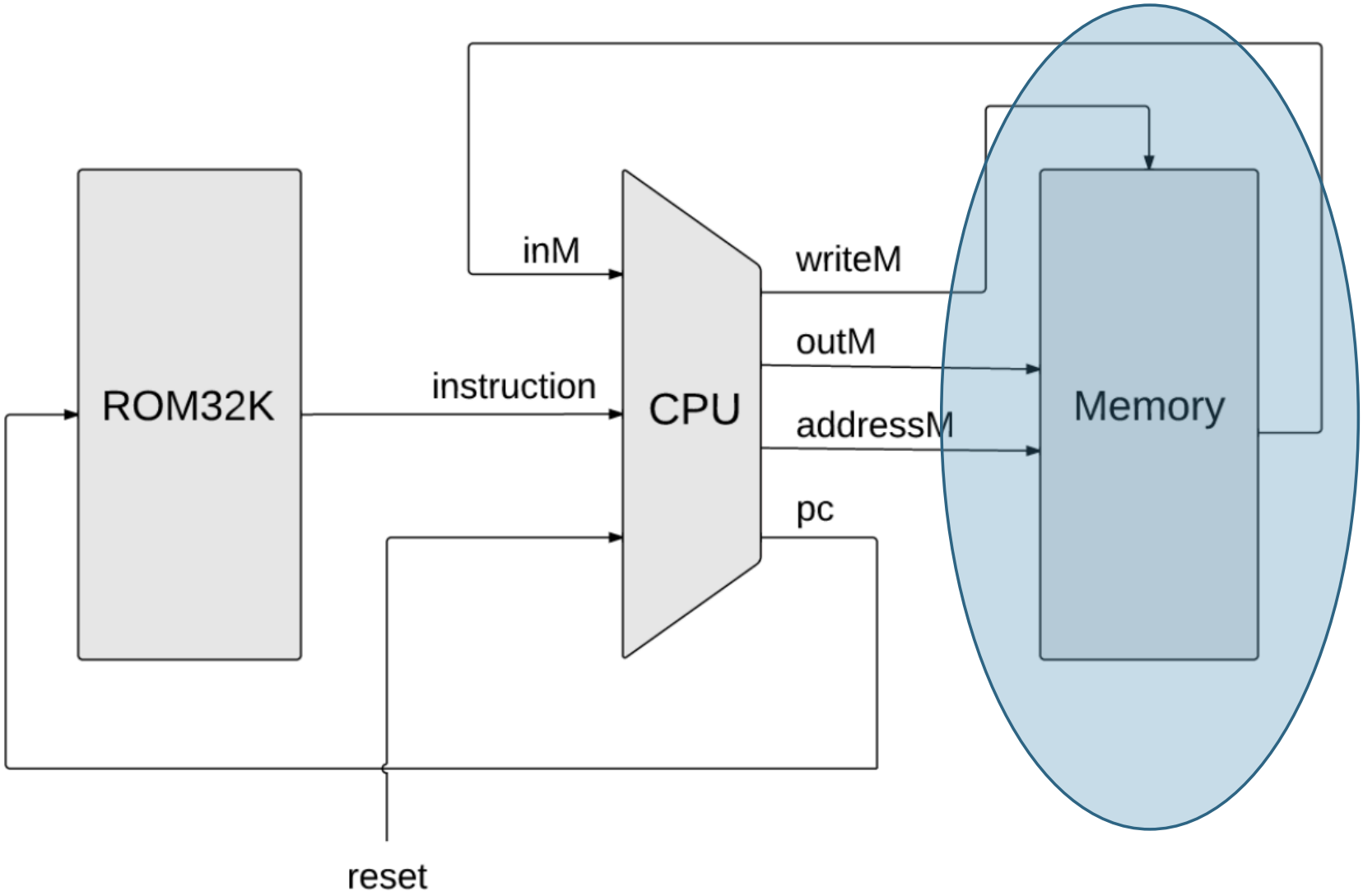

ROM32K
instruction
CPU
inM
writeM
outM
addressM
pc
Memory
reset
Hello,
world

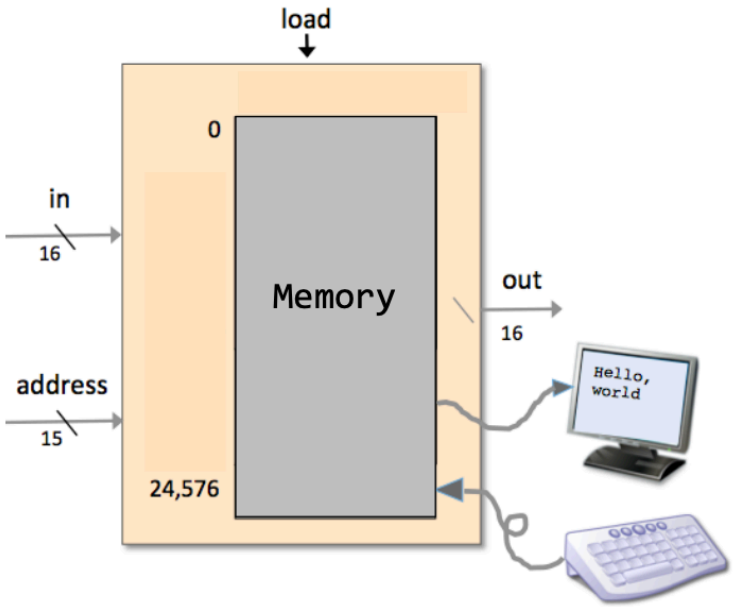

instruction
inM
reset
Hack CPU
outM
writeM
addressM
PC


ALU output
ALU output
c
c's
c
D register
Mux16
A register
instruction
ALU
outM
111 a cccccc ddd jjj
c
inM
M input
Mux16
A register
output
c
c's
c
writeM
addressM
memory address output
reset
reset bit
c
PC
PC
Program Counter output

```
/**
* The Central Processing unit (C
* Consists of an ALU and a set o
registers, * designed to fetch a
execute instructions * written i
Hack machine language.
*/

CHIP CPU {
 IN
  inM[16], // value of M = RAM[A]
  instruction[16], //for execution
  reset; // Signals for restart

 OUT
   outM[16],
   writeM,
   addressM[15],
   pc[15];
 PARTS:
   // Put your code here
}
```




ROM32K
instruction
CPU
inM
writeM
outM
addressM
pc
Memory
reset



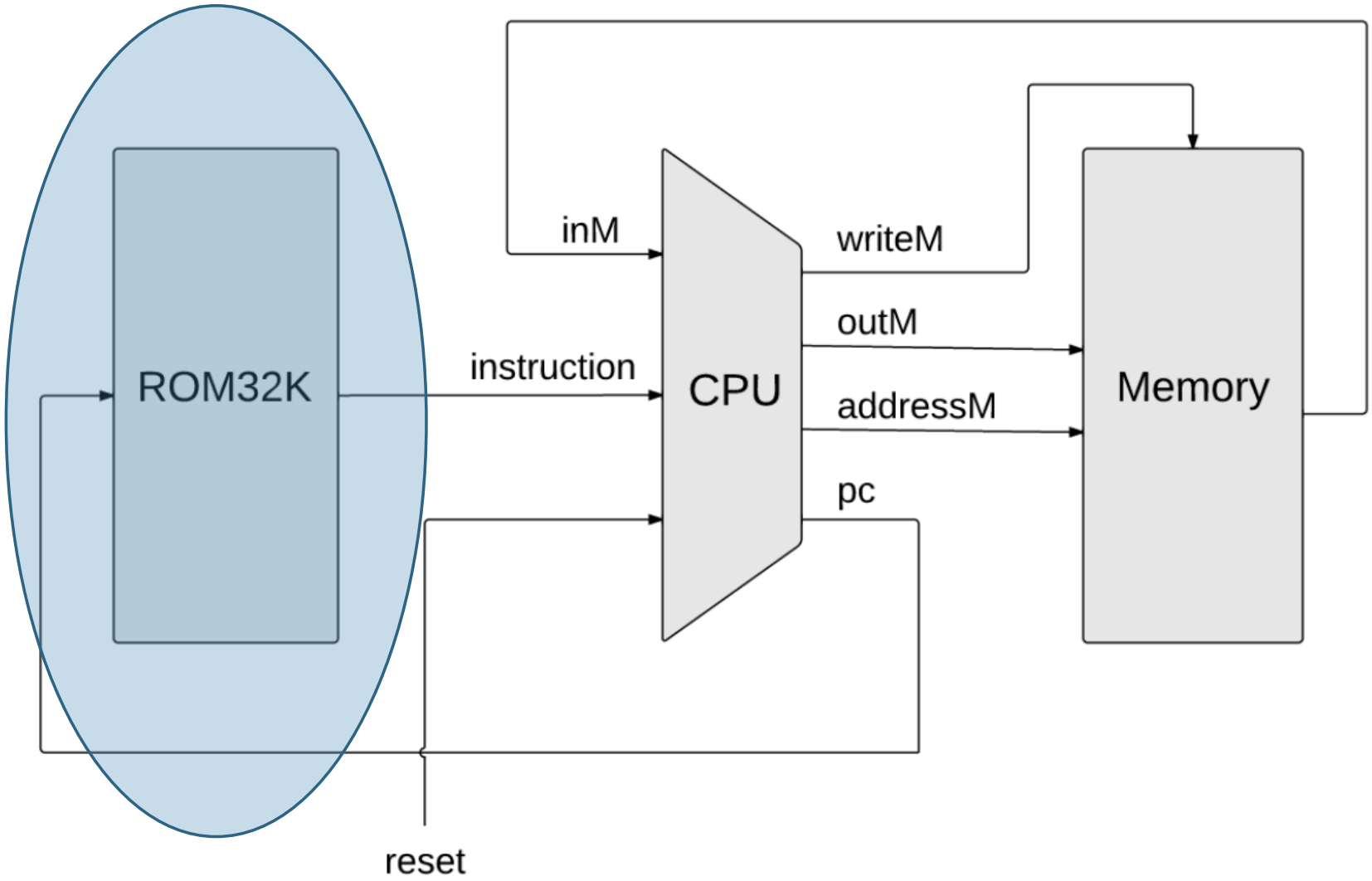

```
/** Memory of 16K 16-bit registers */
CHIP RAM16K {
  IN
    address[14],
}
```

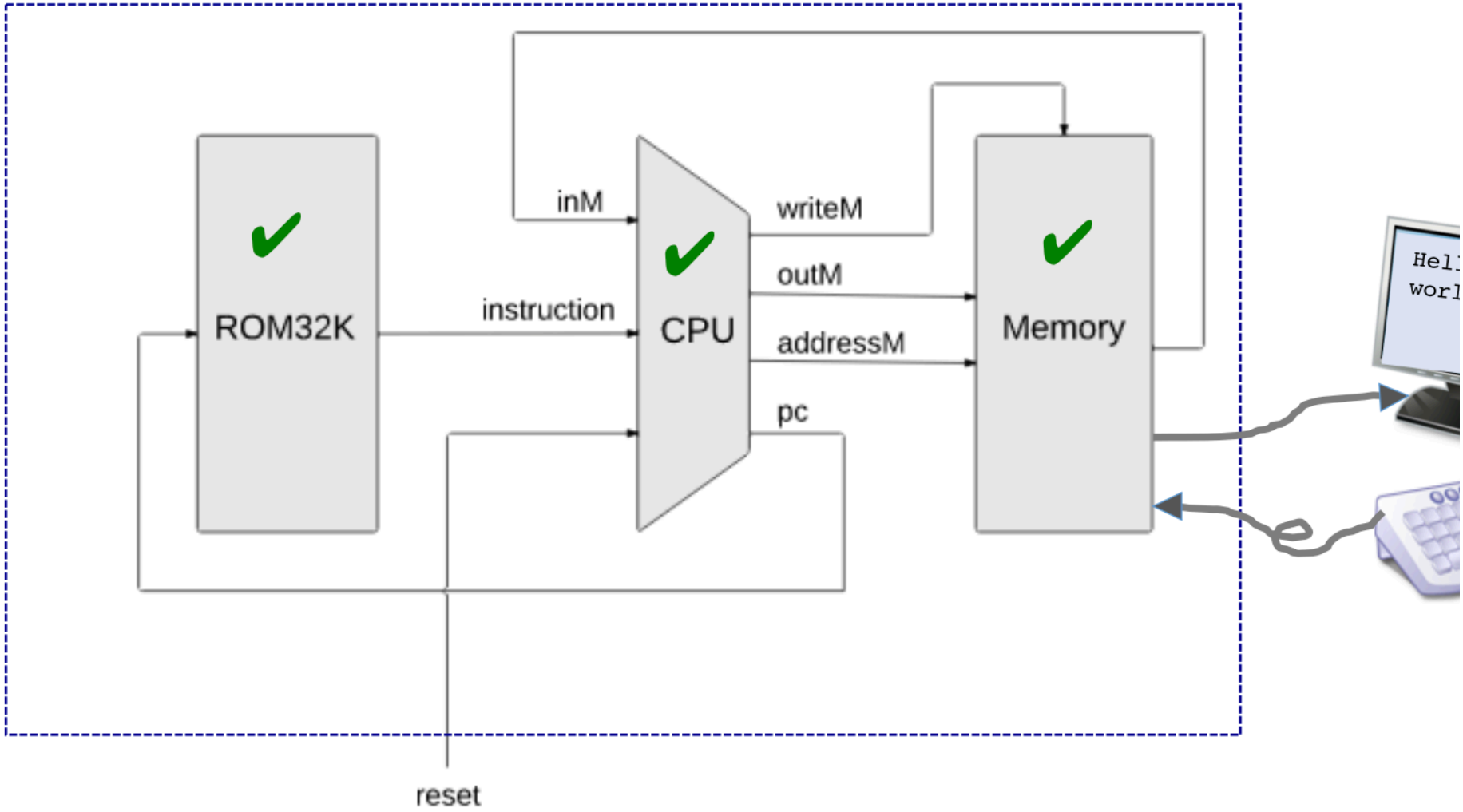
built in project 3

```
/** Memory of 8K 16-bit registers
 *  with a display unit side effect. */
CHIP Screen {
  IN
    address[13],
    in[16],
    load;
  OUT
    out[16];

    BUILTIN
    CLOCKED
}
```

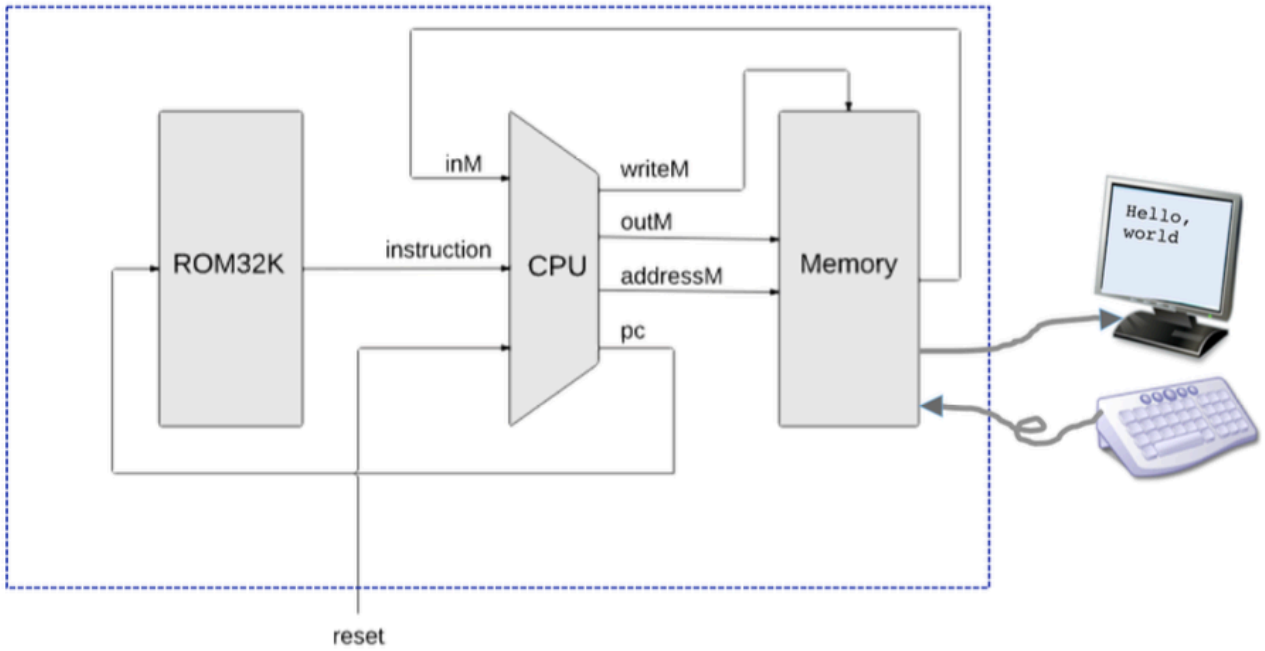
built-in chips

```
/** 16-bit register with a
*  keyboard input side effect */
CHIP Keyboard {
  OUT
    out[16];

    BUILTIN Keyboard;
}
```


ROM32K
inM
instruction
CPU
writeM
outM
addressM
pc
Memory
reset


ROM32K
CPU
Memory
inM
writeM
outM
addressM
pc
instruction
reset

Computer.hdl




```
CHIP Computer {
    IN reset;

    PARTS:
      // your code here
}
```

Coming up: W5.5 Project #5

โพรเจ็ค #5