



การจัดองค์การคอมพิวเตอร์

W5.4 แก้ค์คอมพิวเตอร์

31110321 Computer Organization

สำหรับนักศึกษาชั้นปีที่ 3 สาขาวิชา工กรรมคอมพิวเตอร์

กรุงฤทธิ์ กิติครีว์รพันธุ์

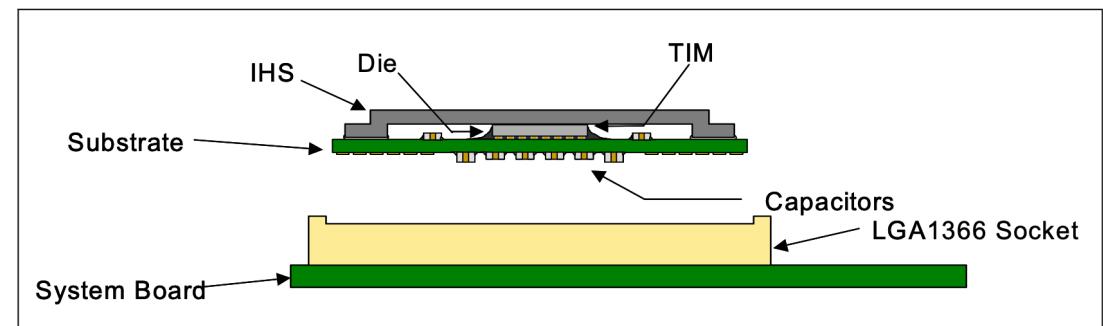
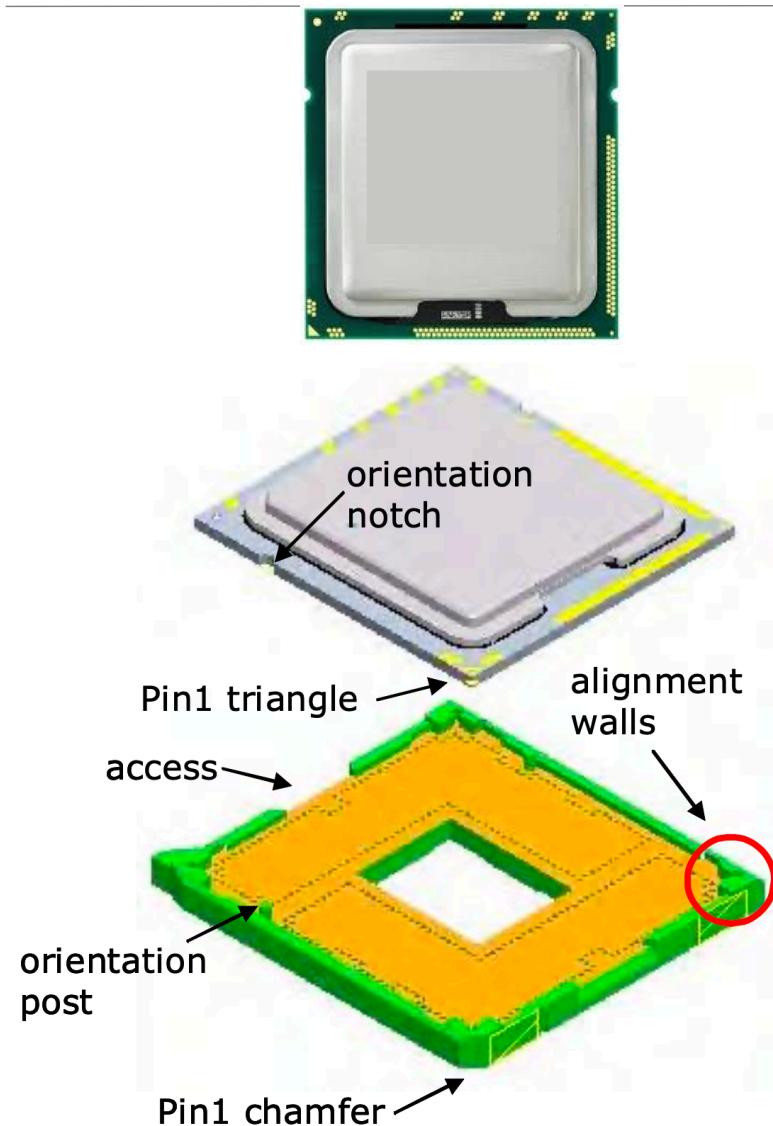
songrit@npu.ac.th

สาขาวิชา工กรรมคอมพิวเตอร์
มหาวิทยาลัยนครพนม

Lecture plan

- 5.1 สถาปัตยกรรมฟอนนอยมันน์
- 5.2 Fetch-Execute Cycle
- 5.3 ชีพิญແວກຄໍ
- **5.4 ແວກຄໍຄວມພິວເຕອນ**
- 5.5 ກາພຣວມໂປຣເຈັກ 5

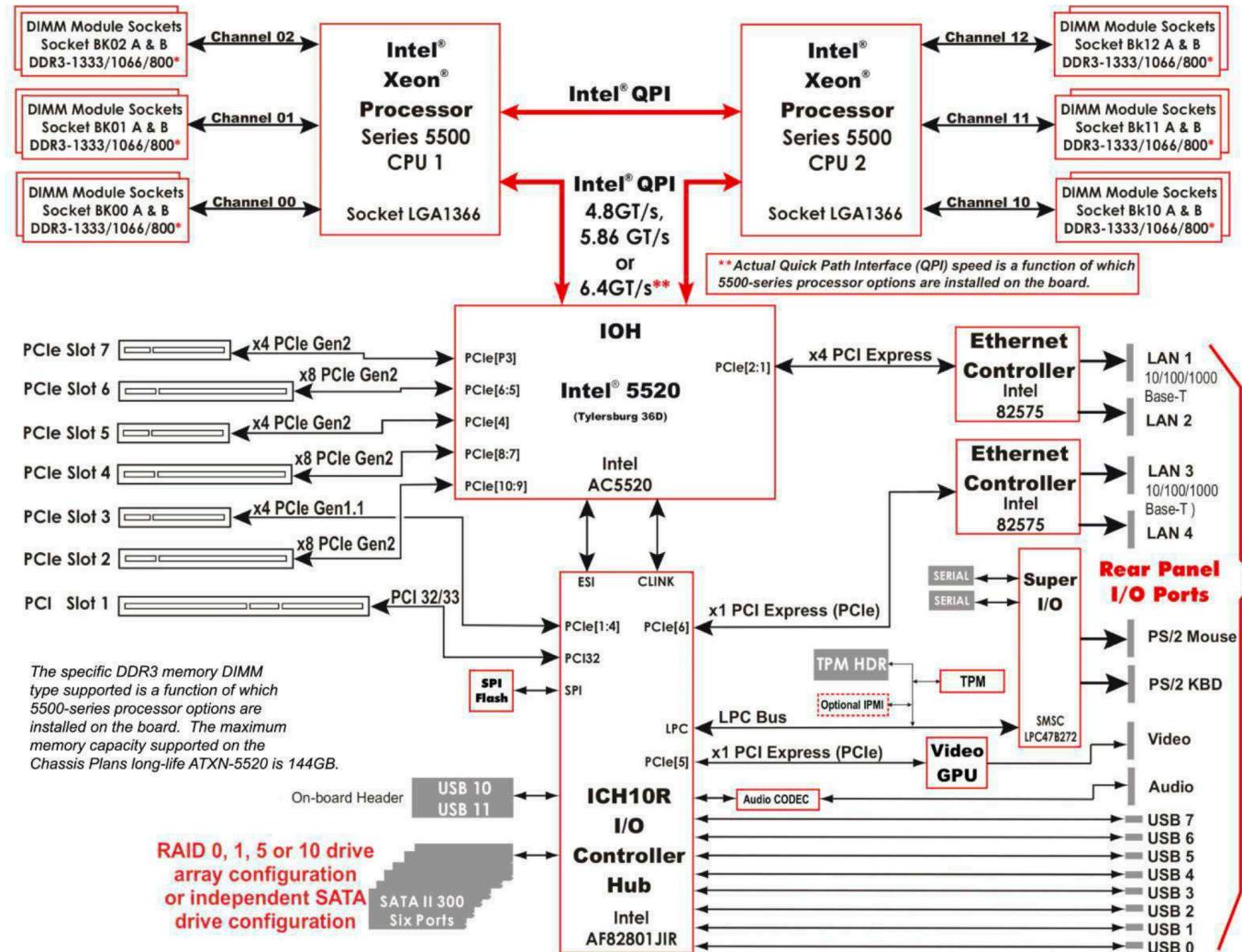
CPU interface (physical)



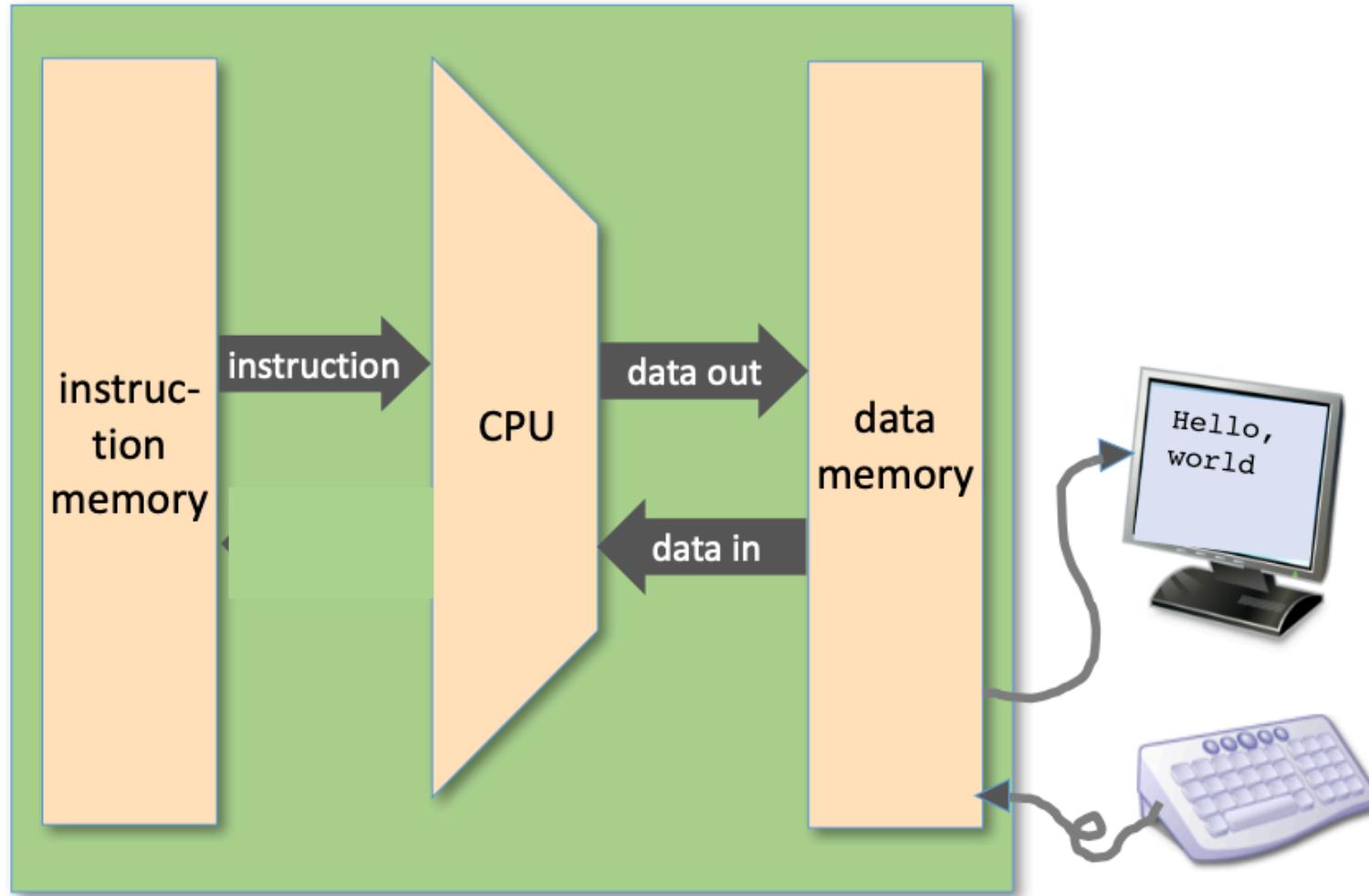
CPU multi processor (logical)



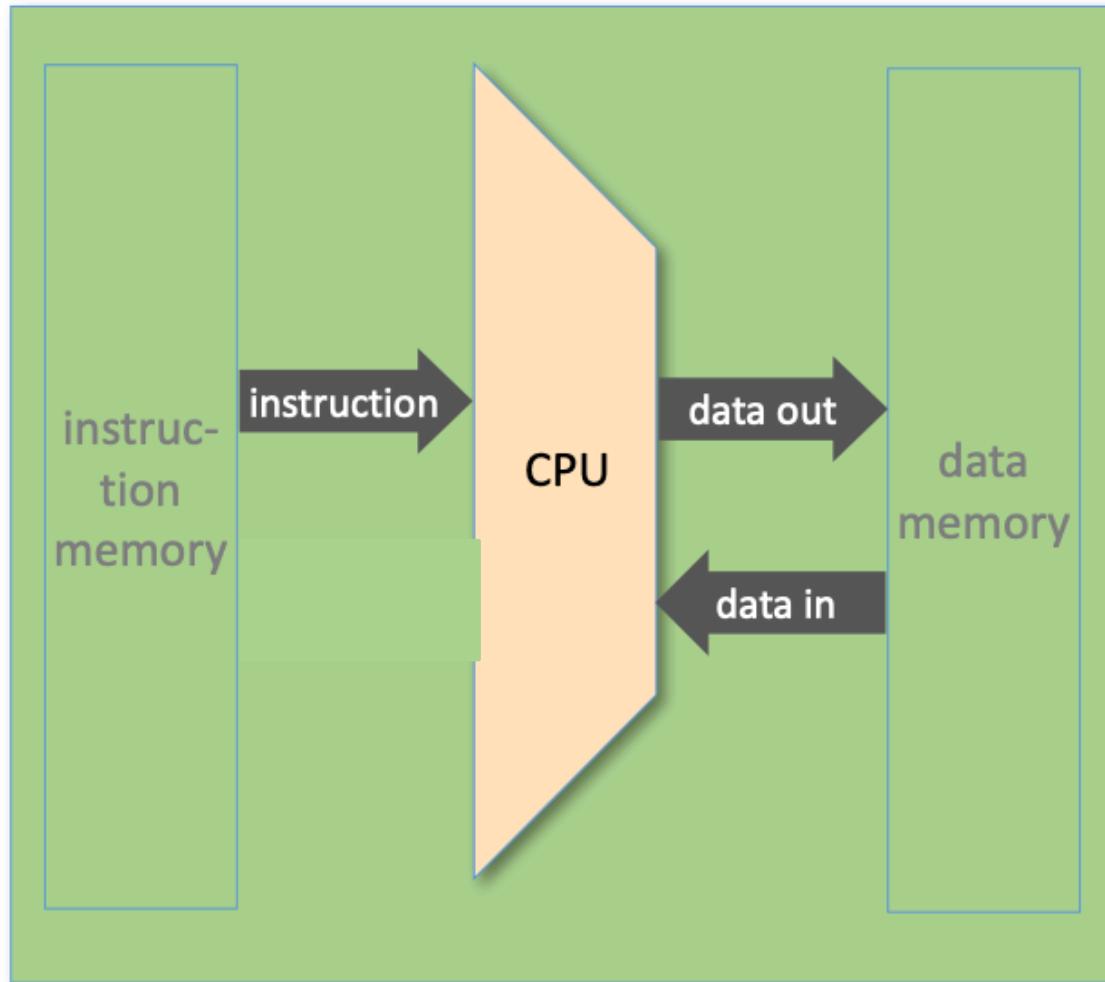
CPU multi processor (logical)



Hack Computer



Hack CPU

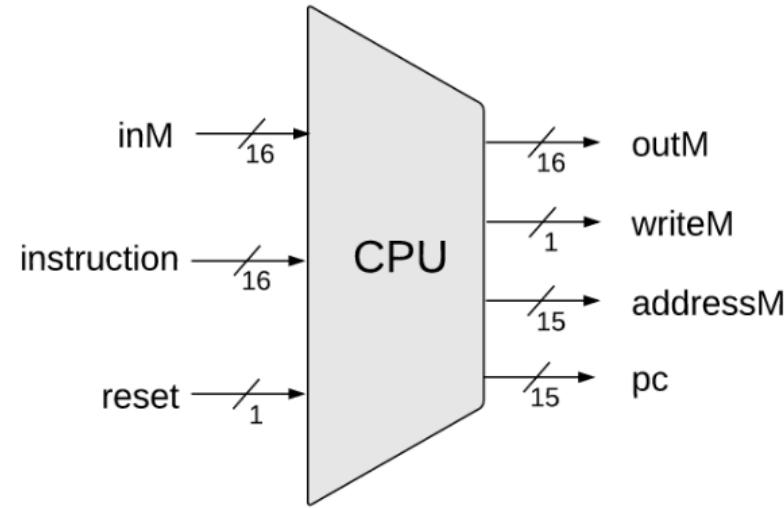


Hack CPU Operation

D=D-A

@17

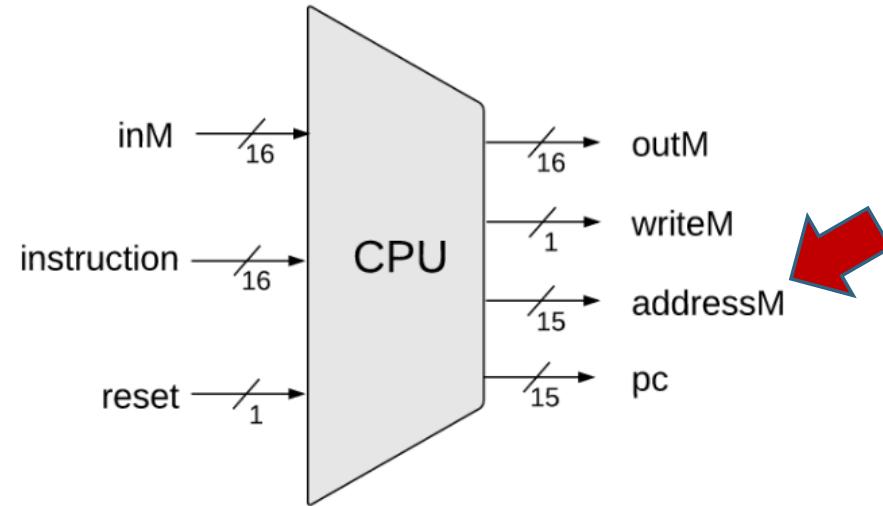
M=M+1



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

Hack CPU Operation

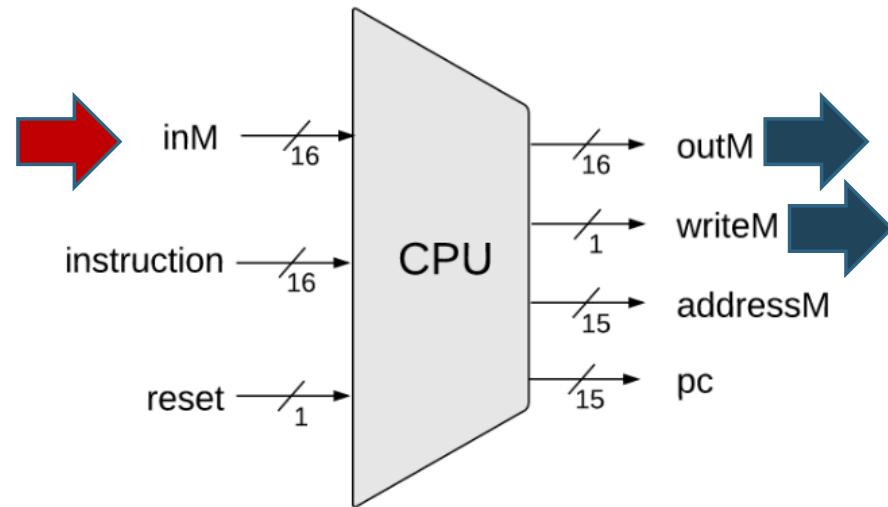
@17



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

Hack CPU Operation

M=M+1

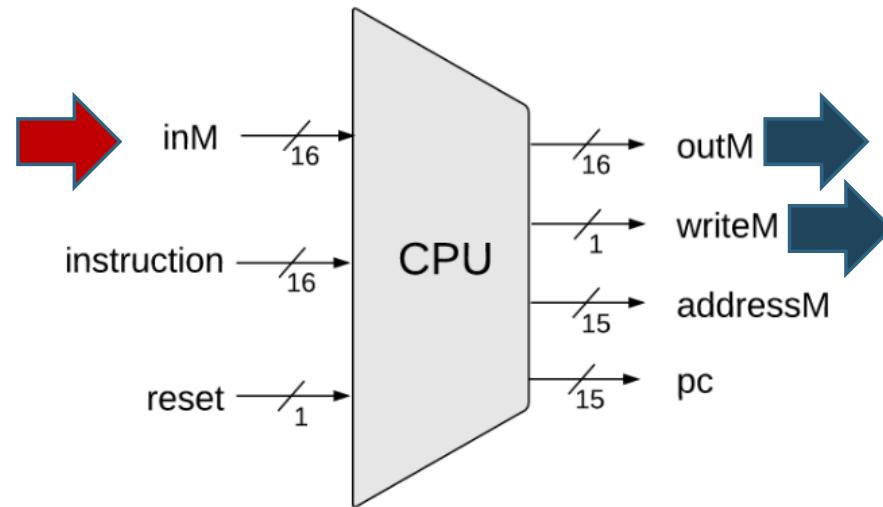


- ถ้าขาเมื่อ(RHS) มีบีโภบค M ซึ่งพิจัยค่าอ่าน inM
- ขาอยเมื่อ(LHS) มีบีโภบค M
 - ส่งออก outM
 - เชต writeM เป็น 1

Hack CPU Operation : Jump

@100

D=D-1; JEQ



- ARegister เก็บ 100

If (reset==0)

เป็นภาวะรันโปรแกรมปกติ CPU พิจารณาว่าจะ
Jump หรือไม่

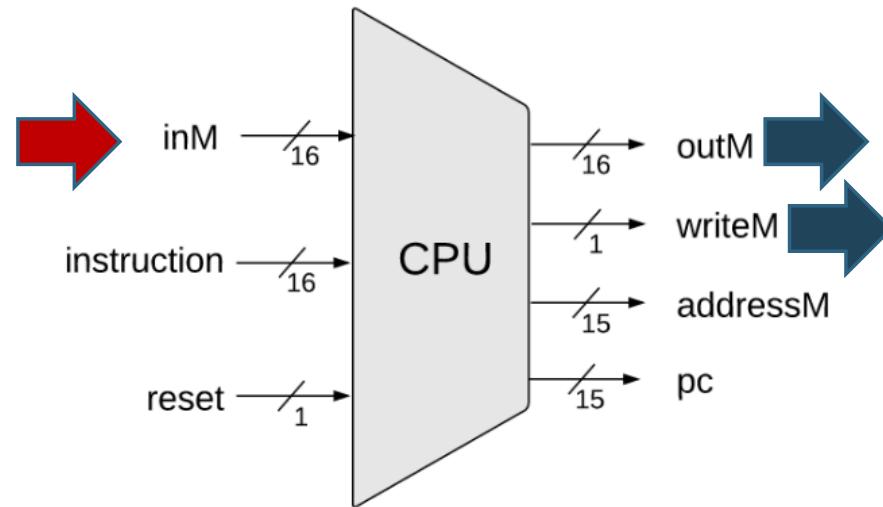
If Jump: เช็ต PC=A

Else: PC++

Hack CPU Operation : Jump

@100

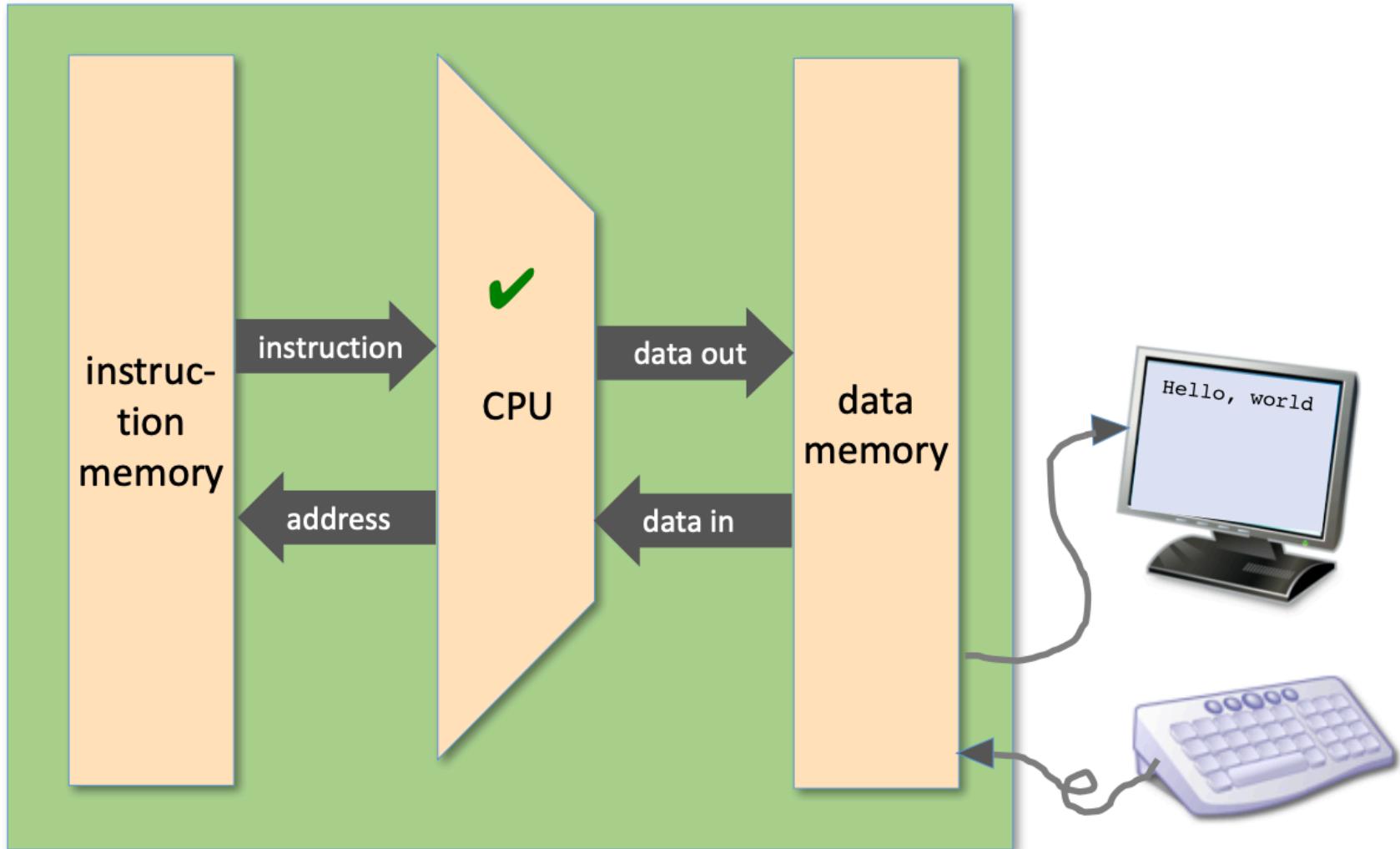
D=D-1; JEQ



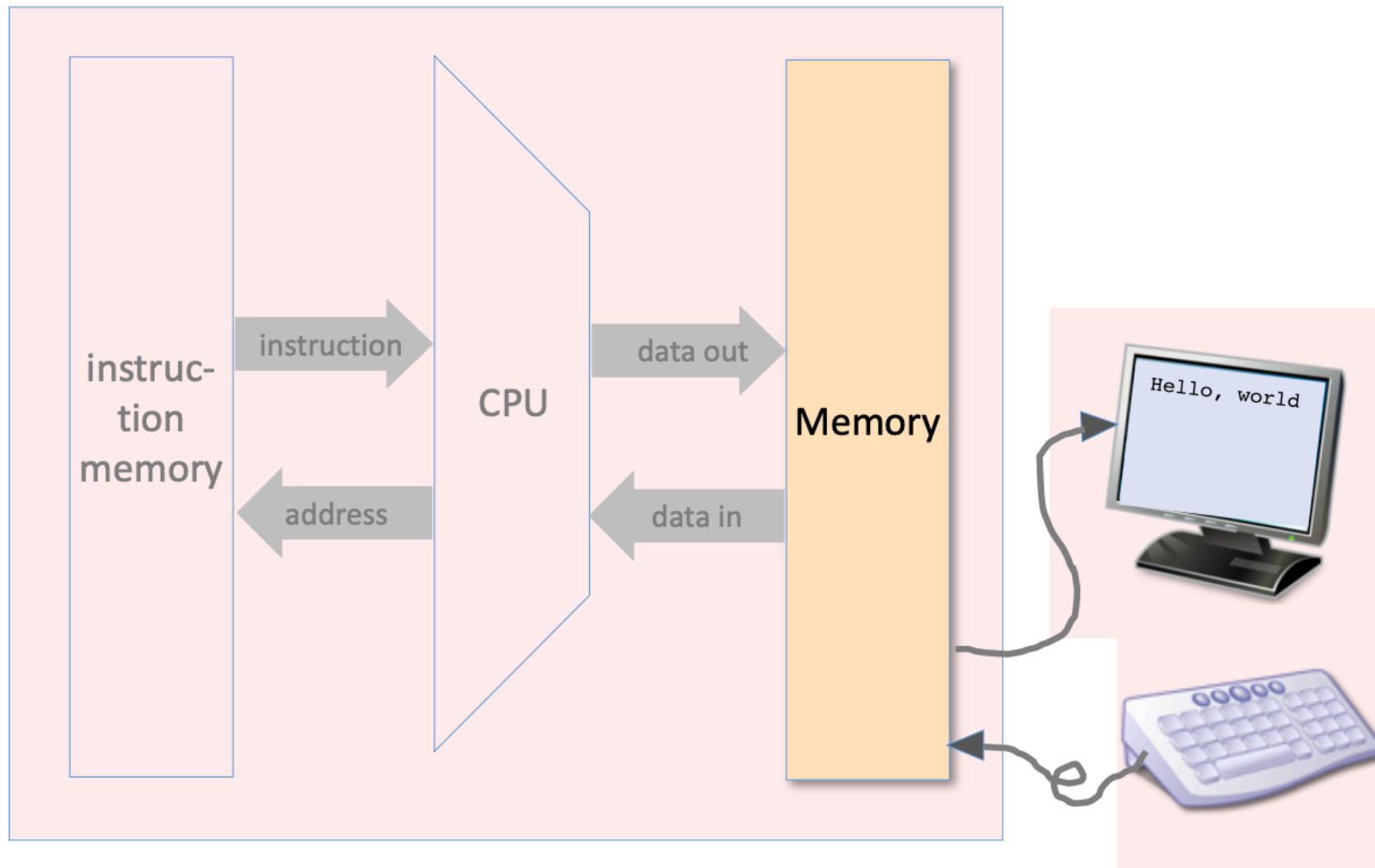
If (reset==1)

หมายถึง user กดปุ่ม reset ต้องการเริ่ม
โปรแกรมใหม่
PC=0

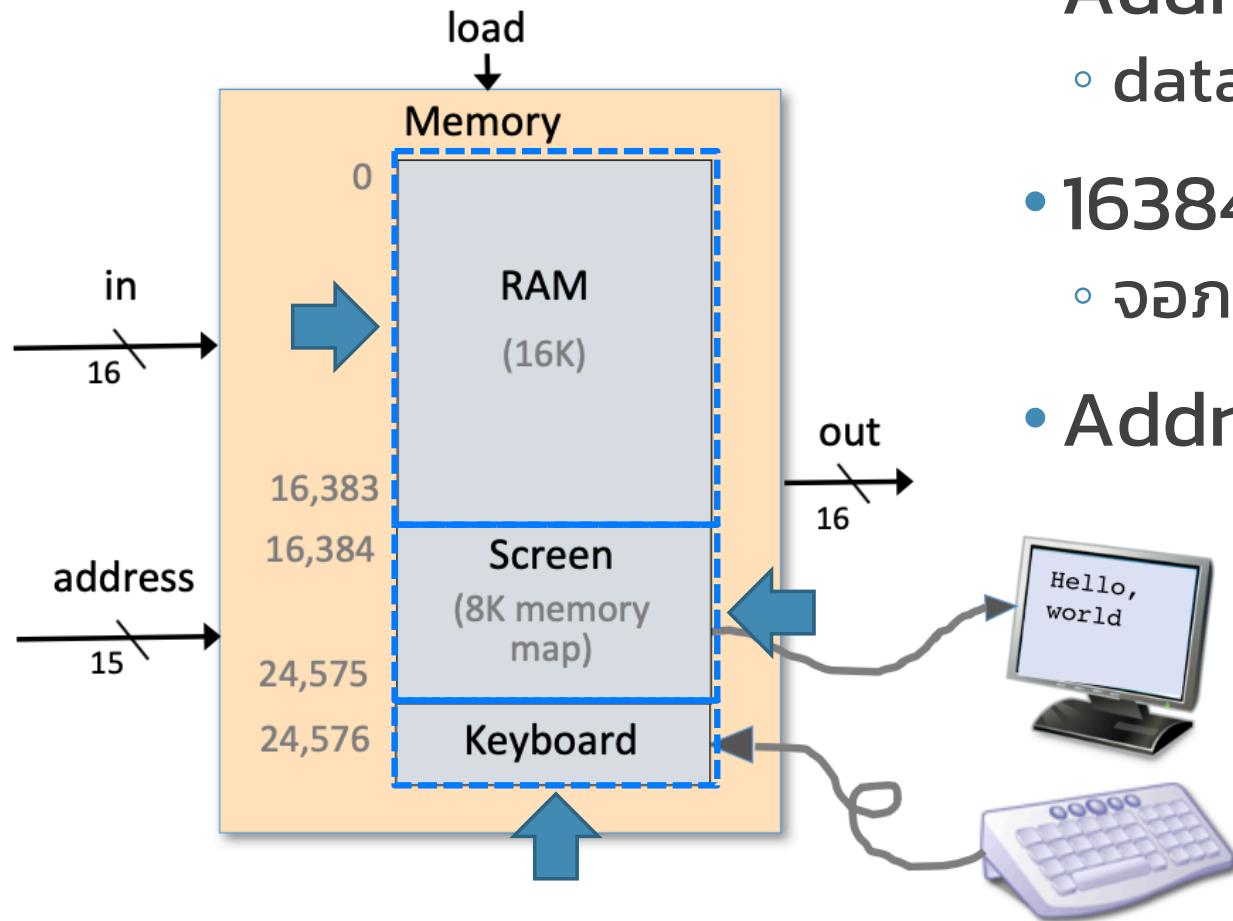
ແອກຄ່ຄວມພິວເຕວ່າ



Memory



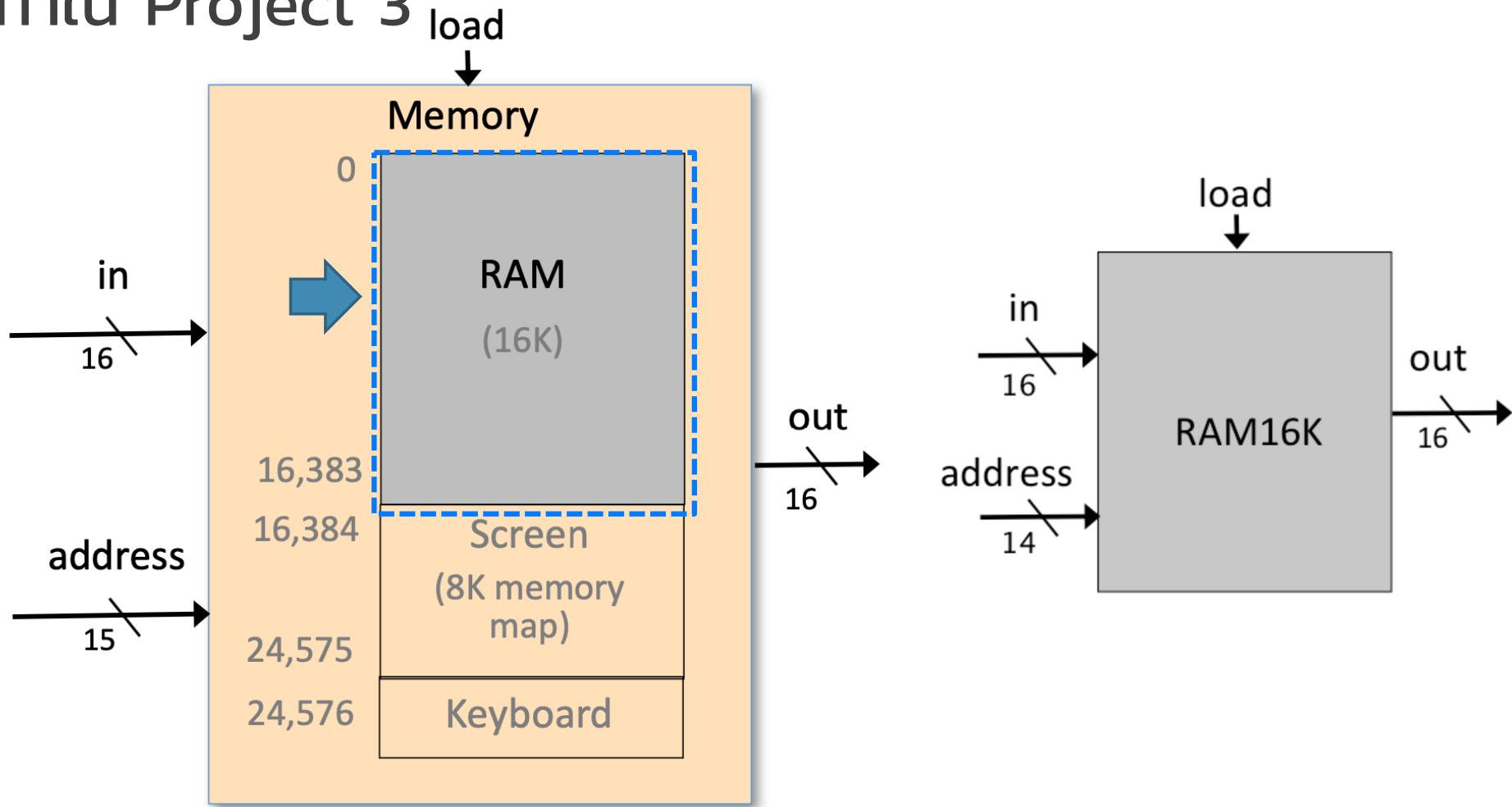
Memory : គណចែប



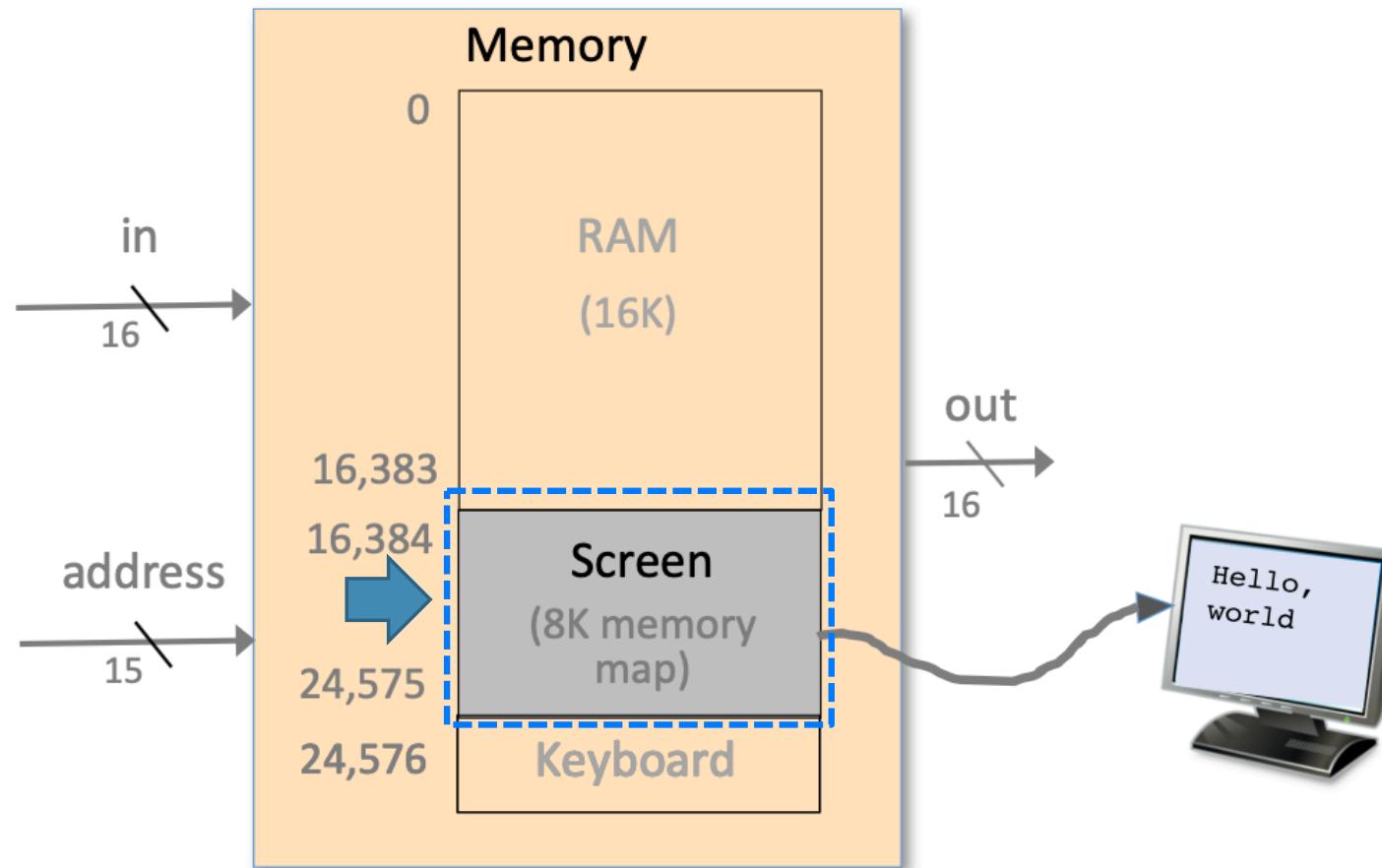
- Address 0 តីង 16383
 - data memory
- 16384 តីង 24575
 - រាជរដ្ឋ
- Address 24576 → គឺបន្ទុក

Memory : implement

- ทำใน Project 3



Screen



Memory mapped output

Screen

(16384)	0	1111010100000000
	1	0000000000000000
	:	
31	0011000000000001	
32	0000101000000000	
33	0000000000000000	
	:	
63	0000000000000000	
	:	
8159	0000000000000000	
8160	0000000000000000	
	:	
8191	1011010100000000	

แก้ว 0

refresh

แก้ว 1

แก้ว 255

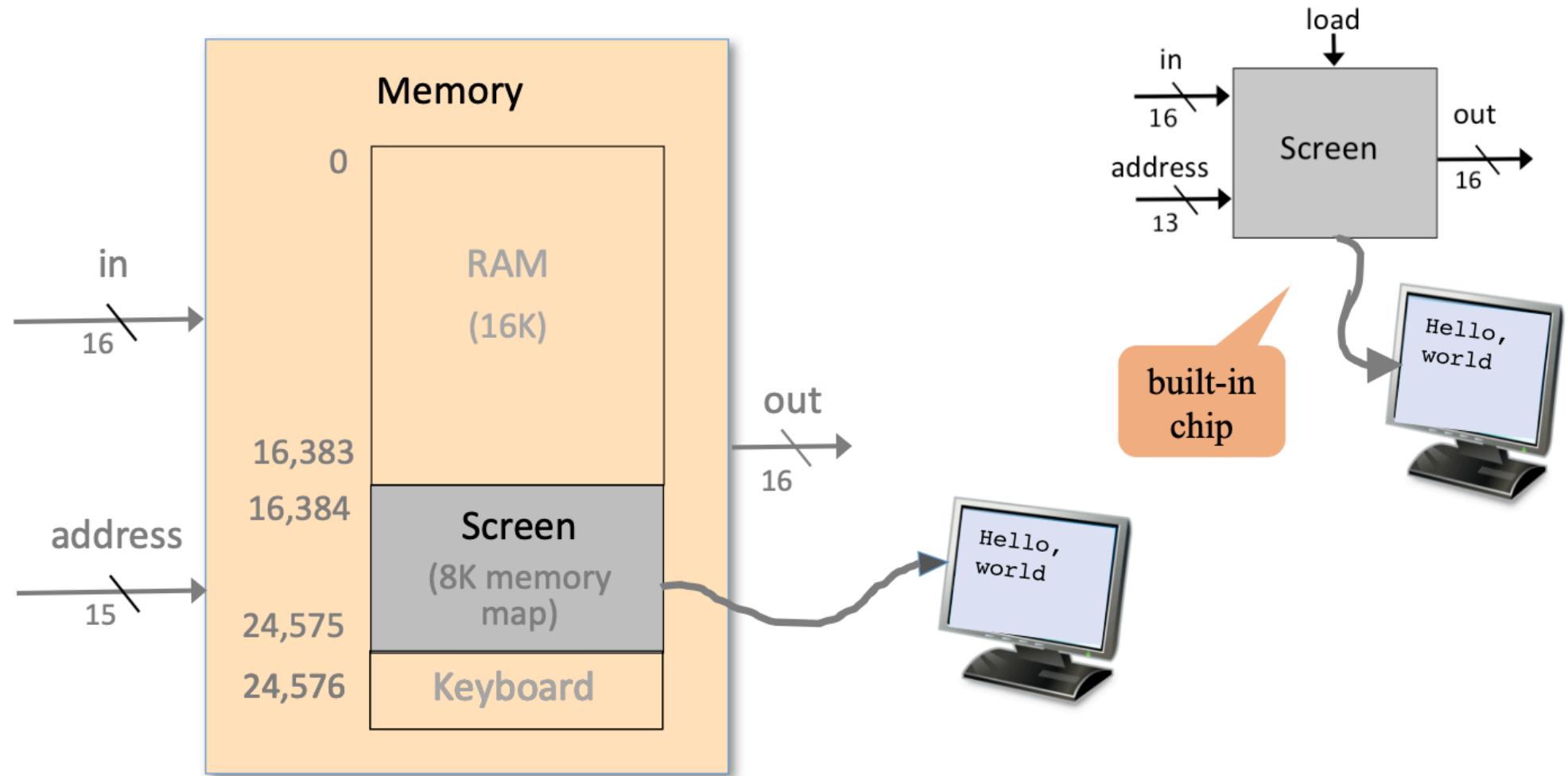
Display Unit (256 by 512, b/w)



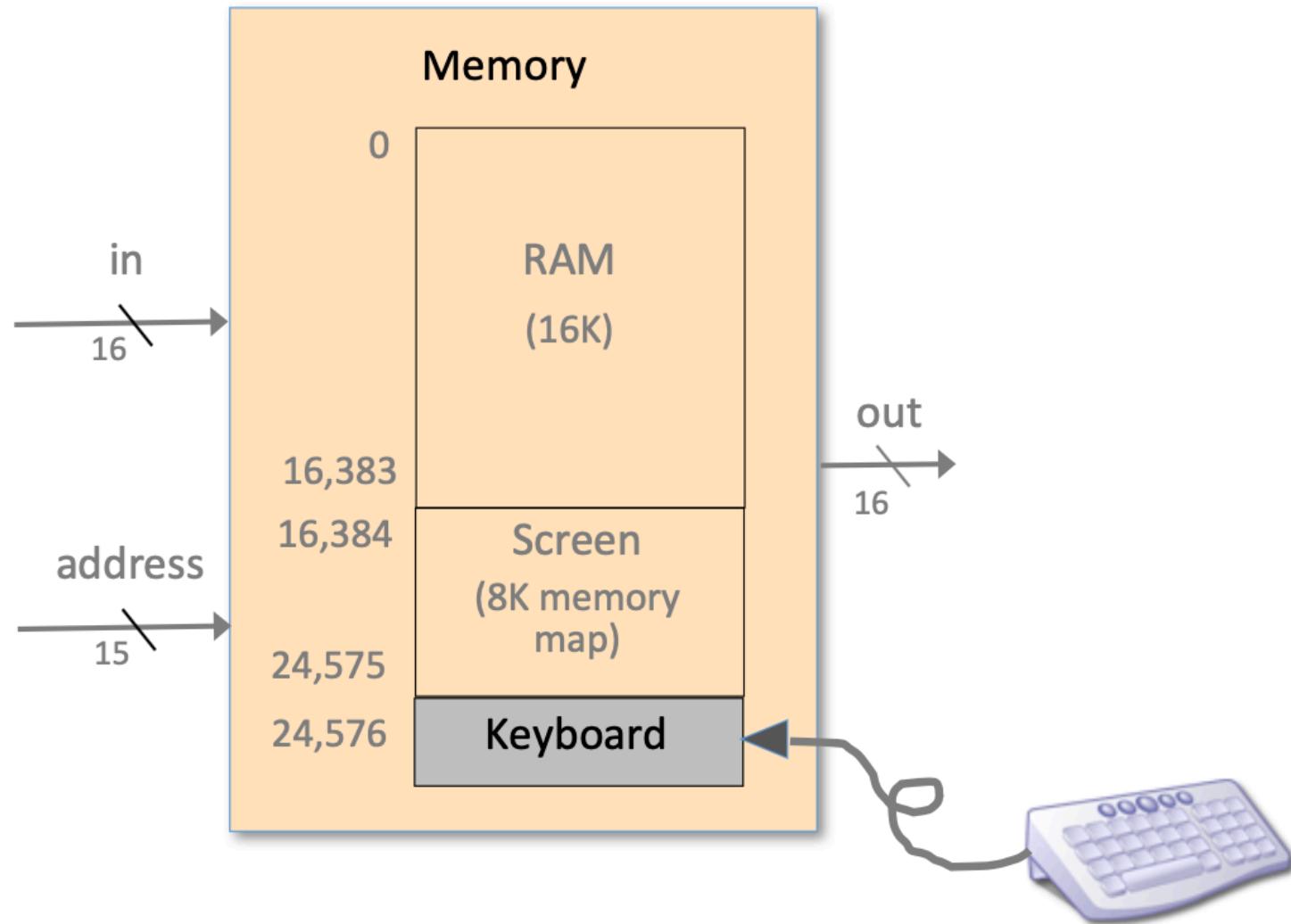
$$\text{RAM}[0] = -1$$

$$= 1111111111111111$$

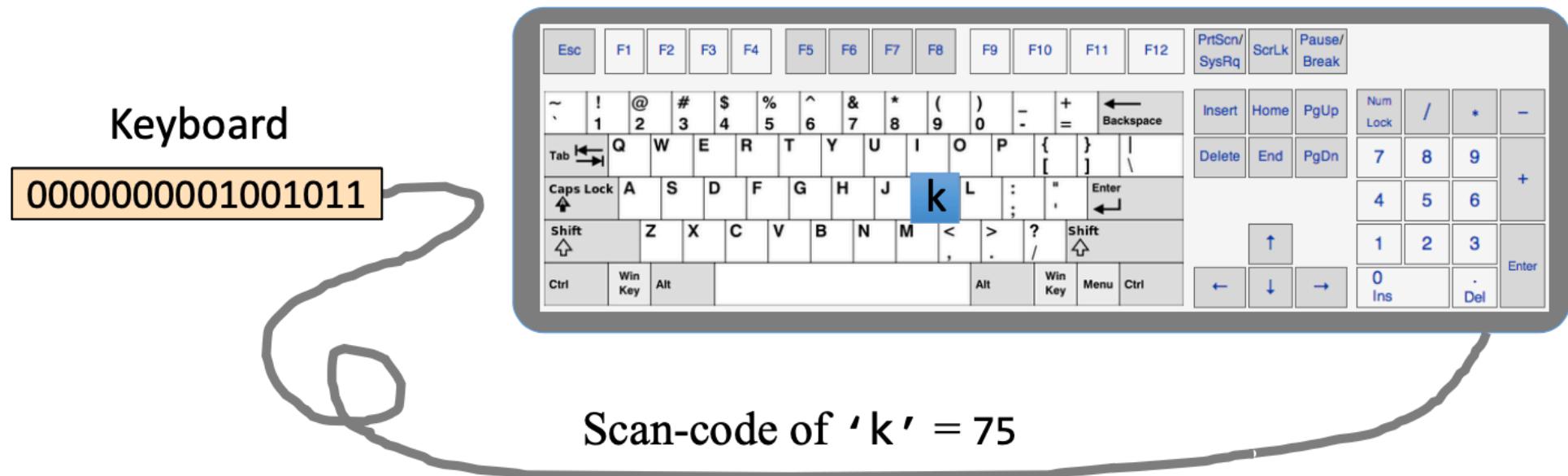
Screen



Keyboard



Keyboard memory map



Keyboard character set

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

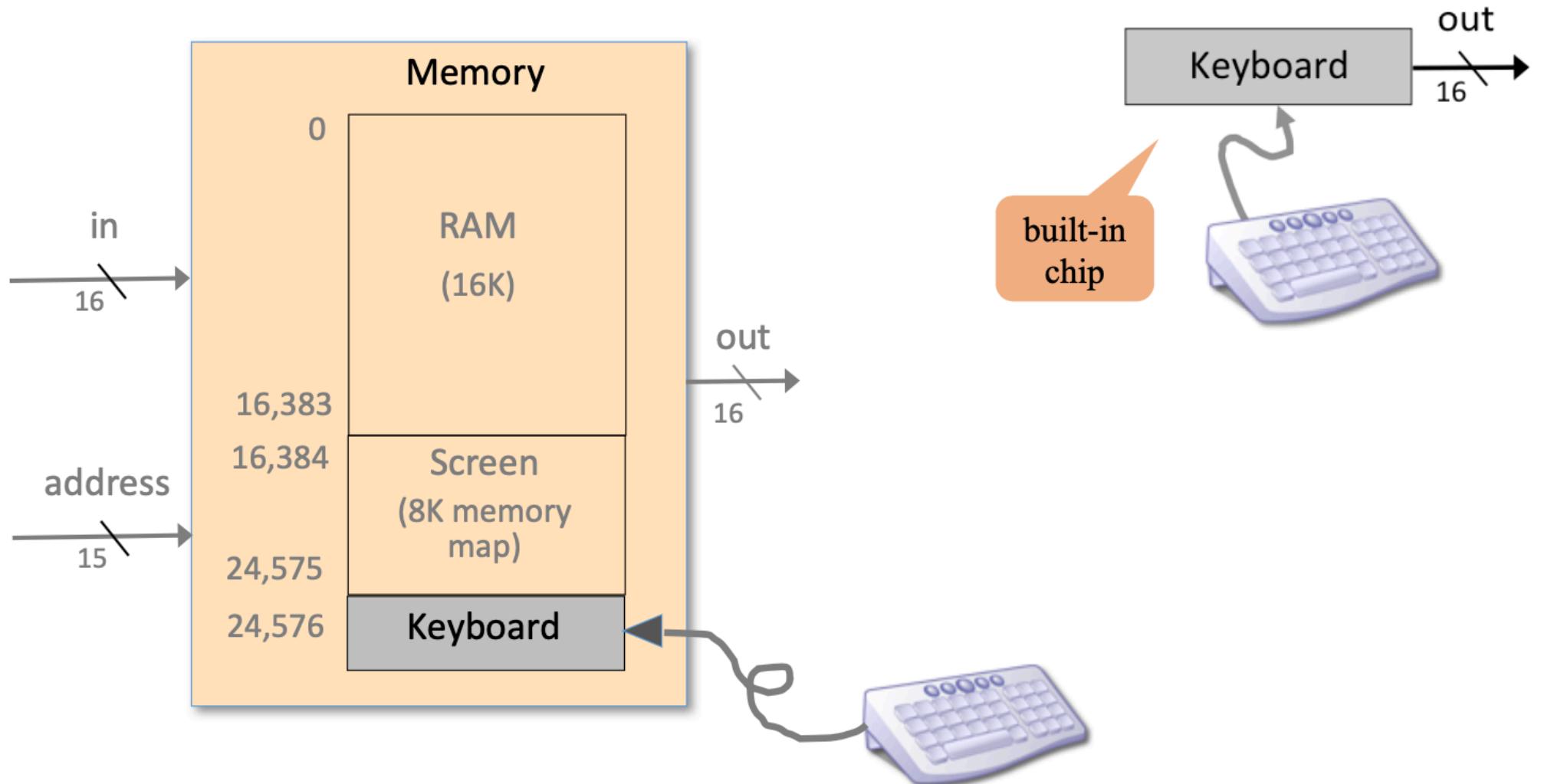
key	code
0	48
1	49
...	...
9	57
:	58
;	59
<	60
=	61
>	62
?	63
@	64

key	code
A	65
B	66
C	...
...	...
Z	90
[91
/	92
]	93
^	94
_	95
`	96

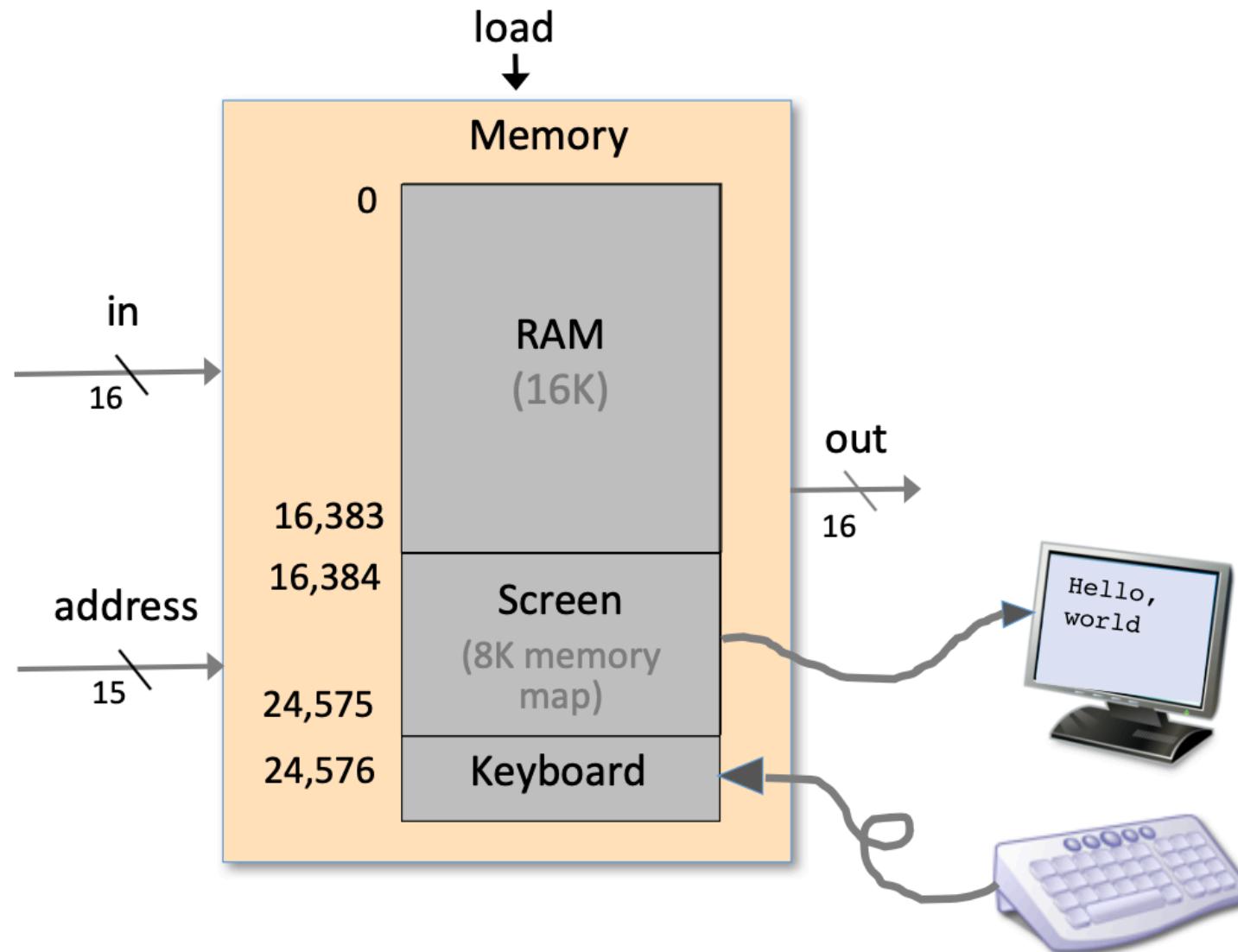
key	code
a	97
b	98
c	99
...	...
z	122
{	123
	124
}	125
~	126

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

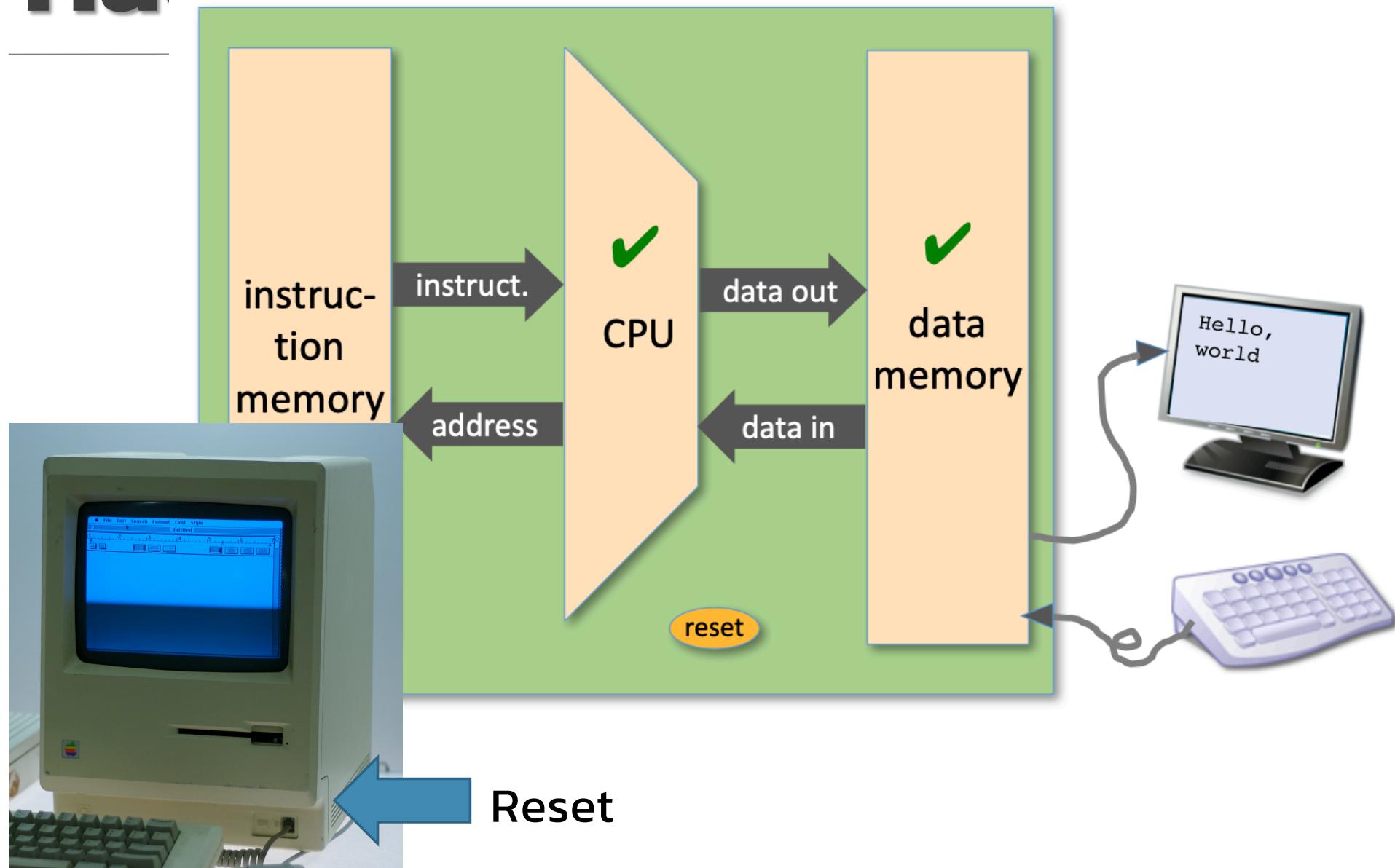
Keyboard



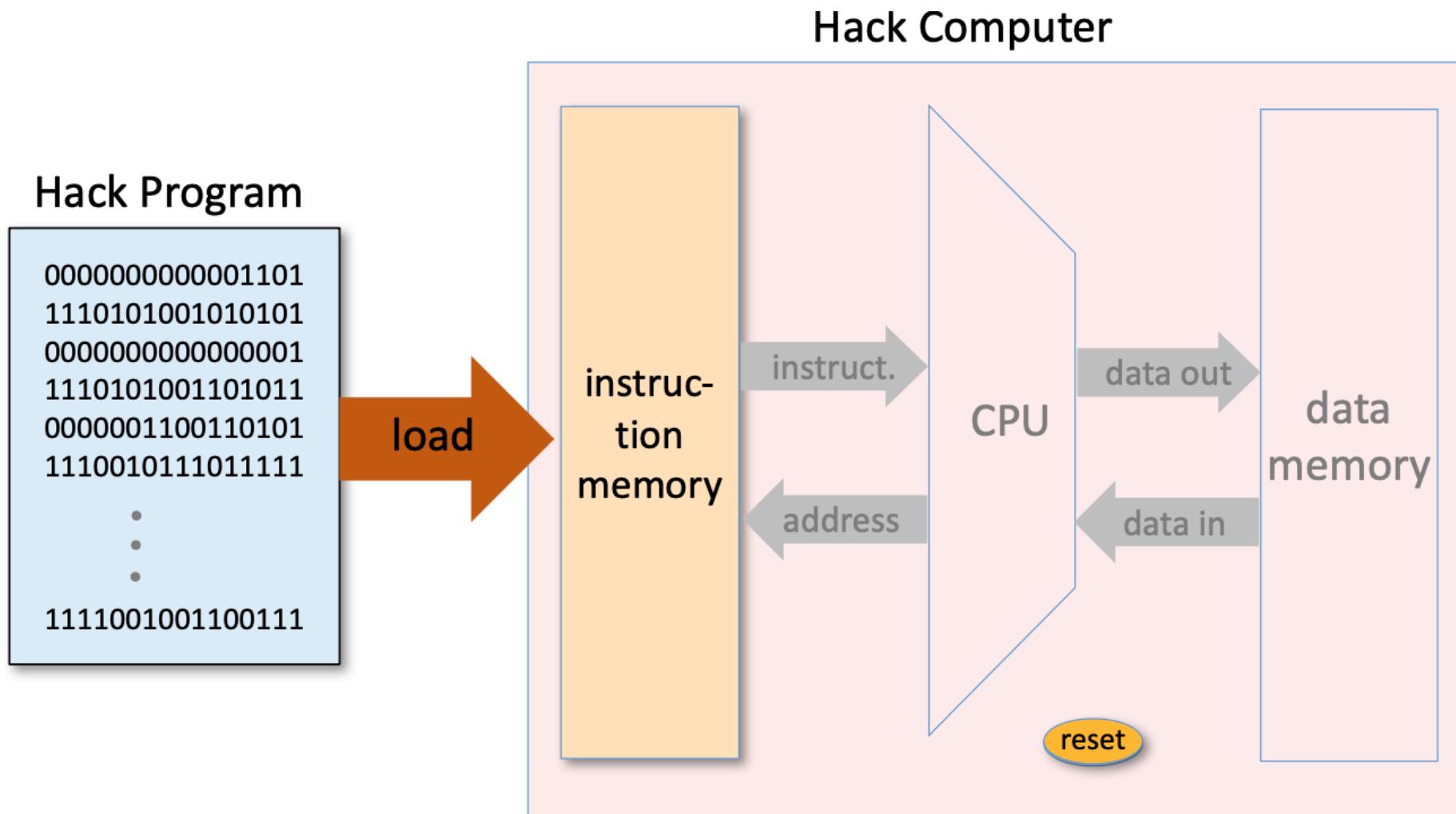
Memory implementation



Hack Computer



Instruction memory

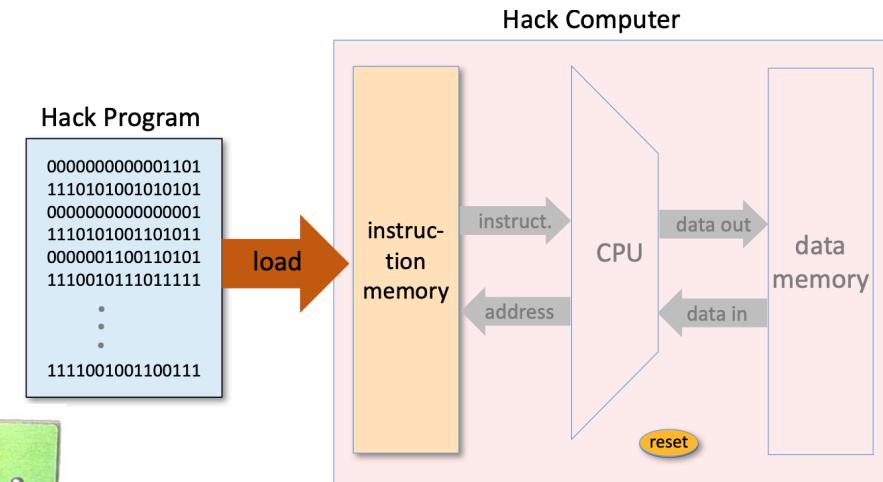


Instruction memory

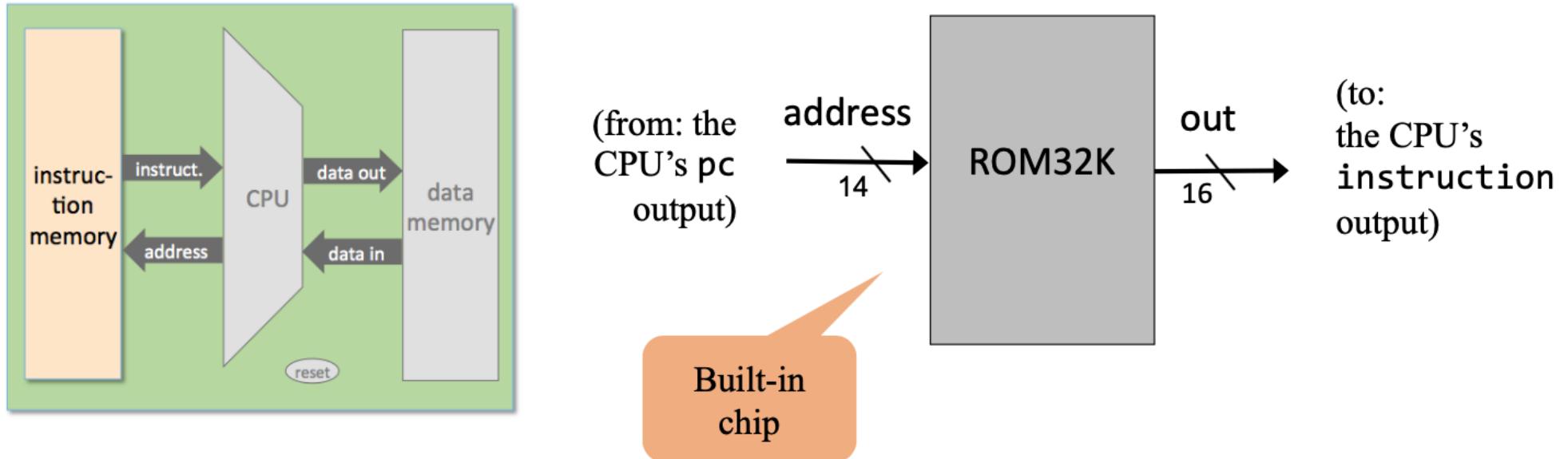
- Hardware implement
 - ใส่ instruction ลง ROM chip



- Hack CPU simulation
 - instruction เก็บเป็น text file

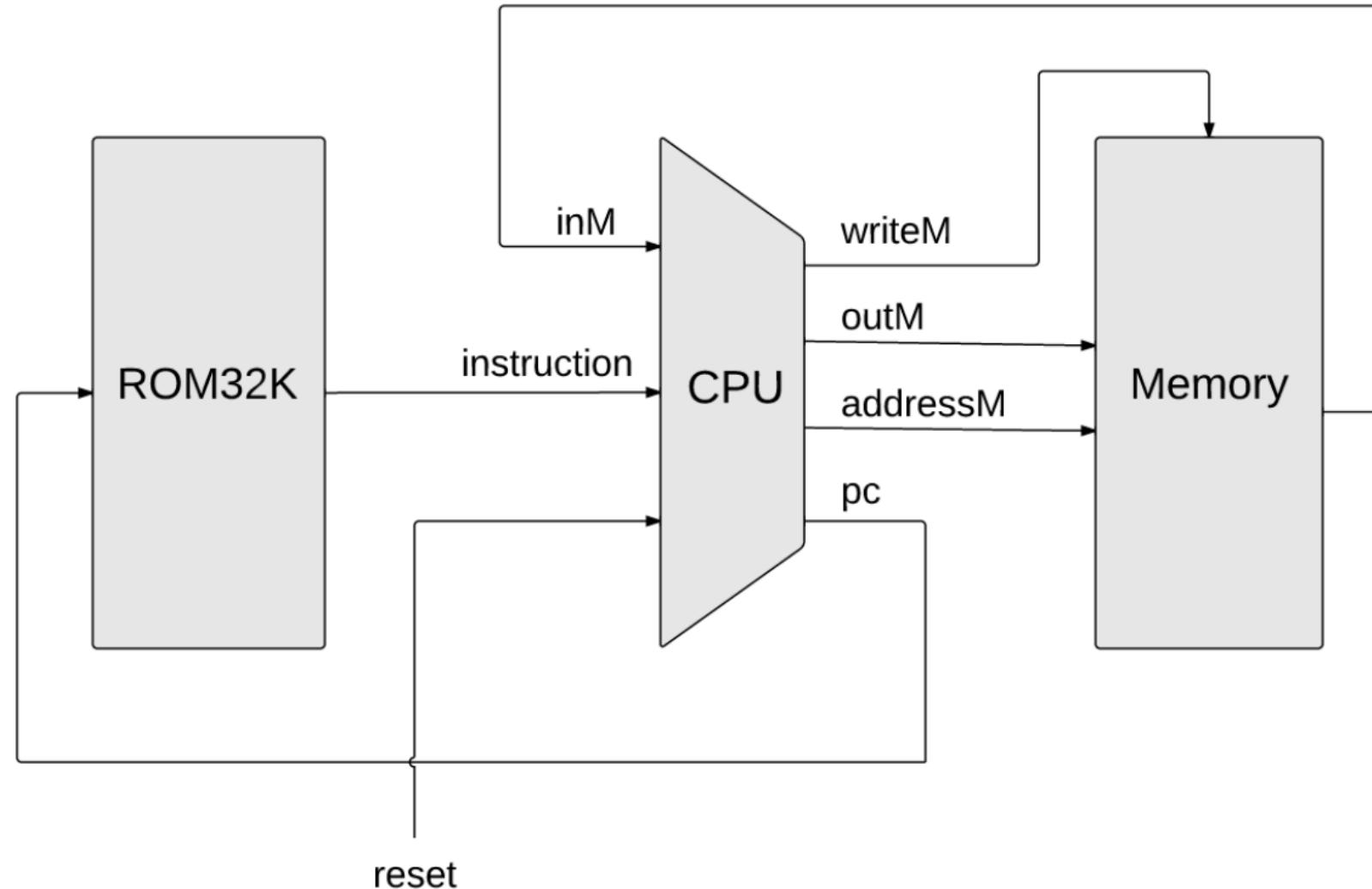


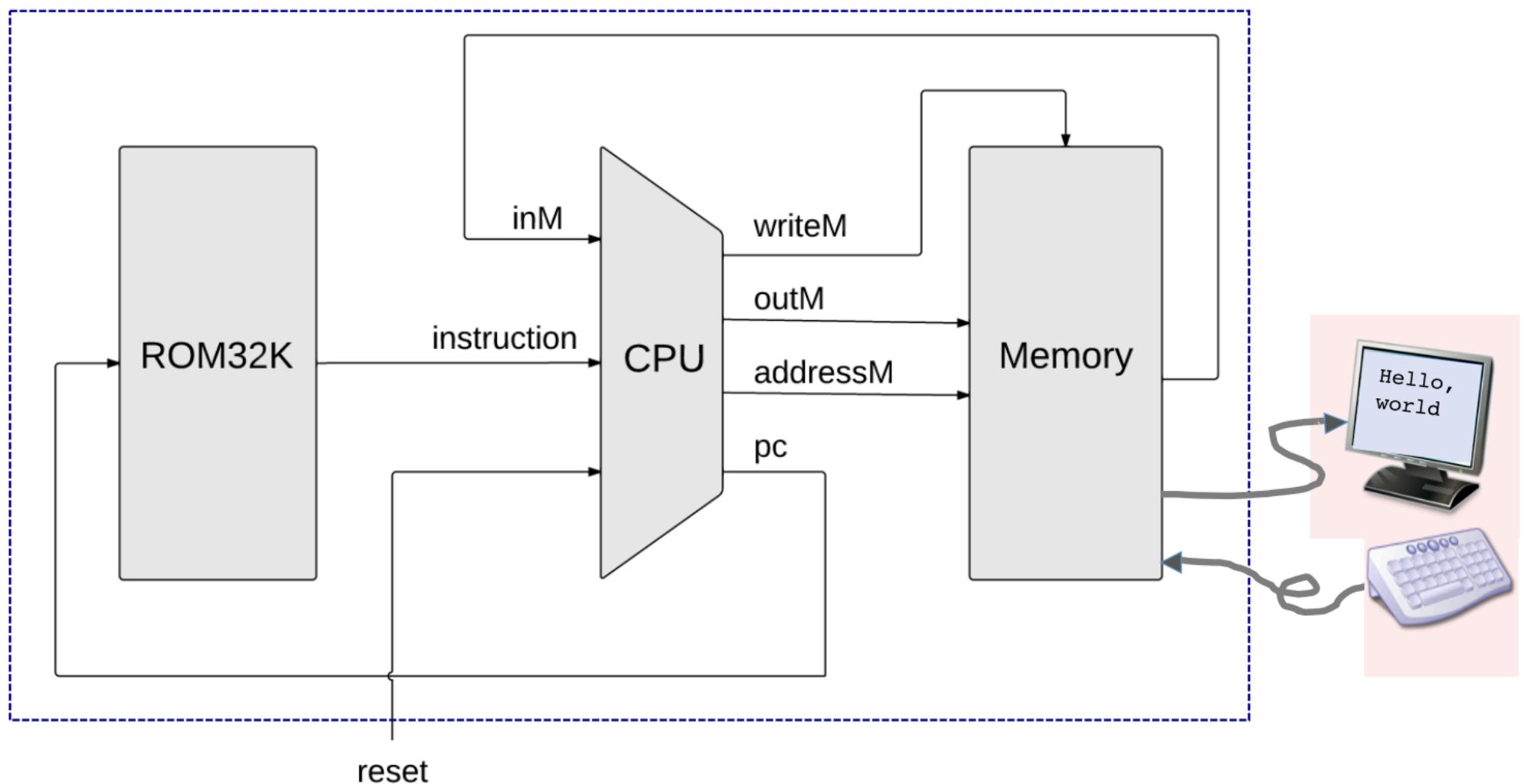
Instruction memory



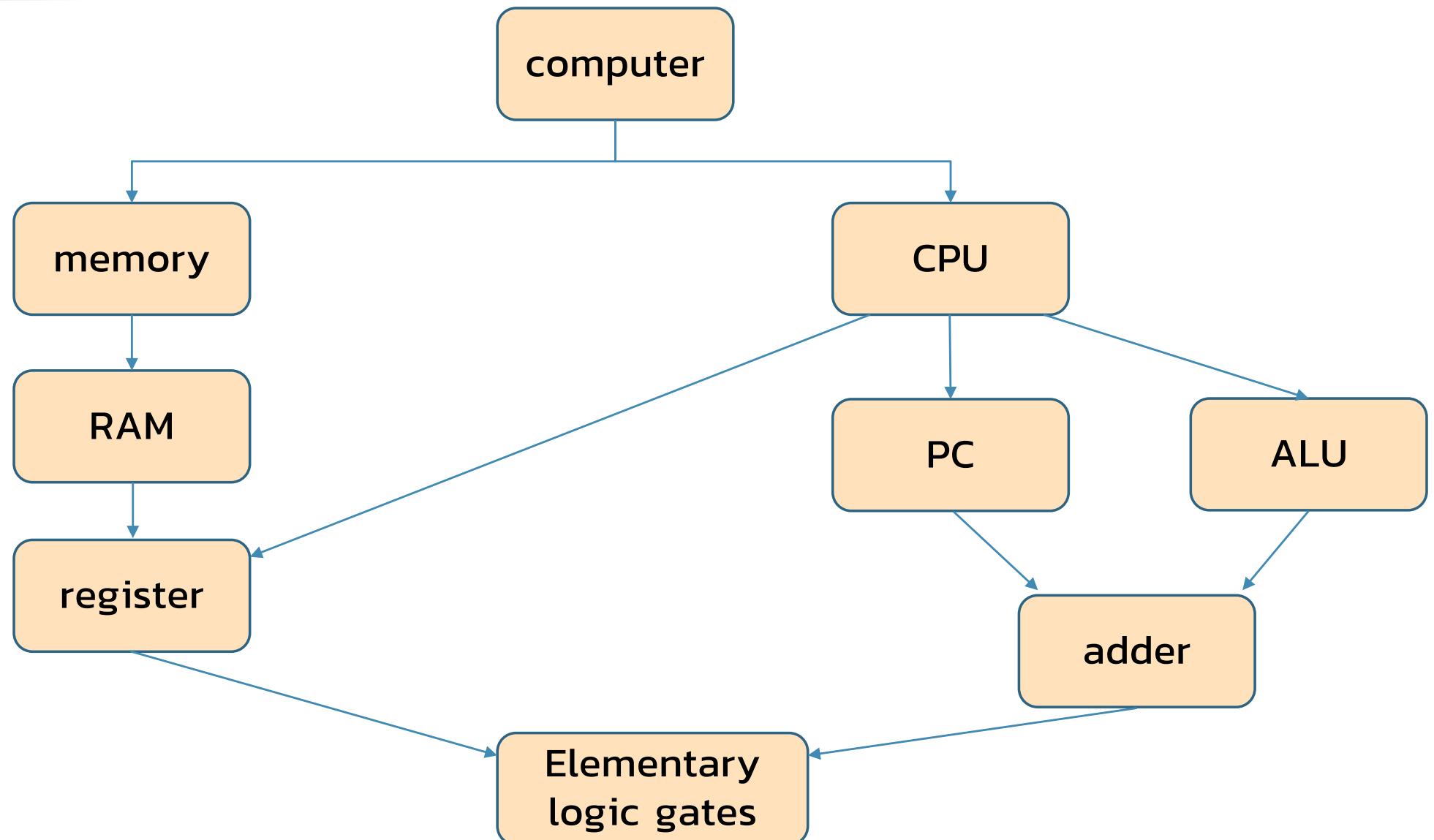
- Hack instruction Memory
 - Build-in chip ឬ ROM32K
 - ROM32K: ឬូវ read-only , 16-bit, 32K RAM chip

Hack Computer implementation

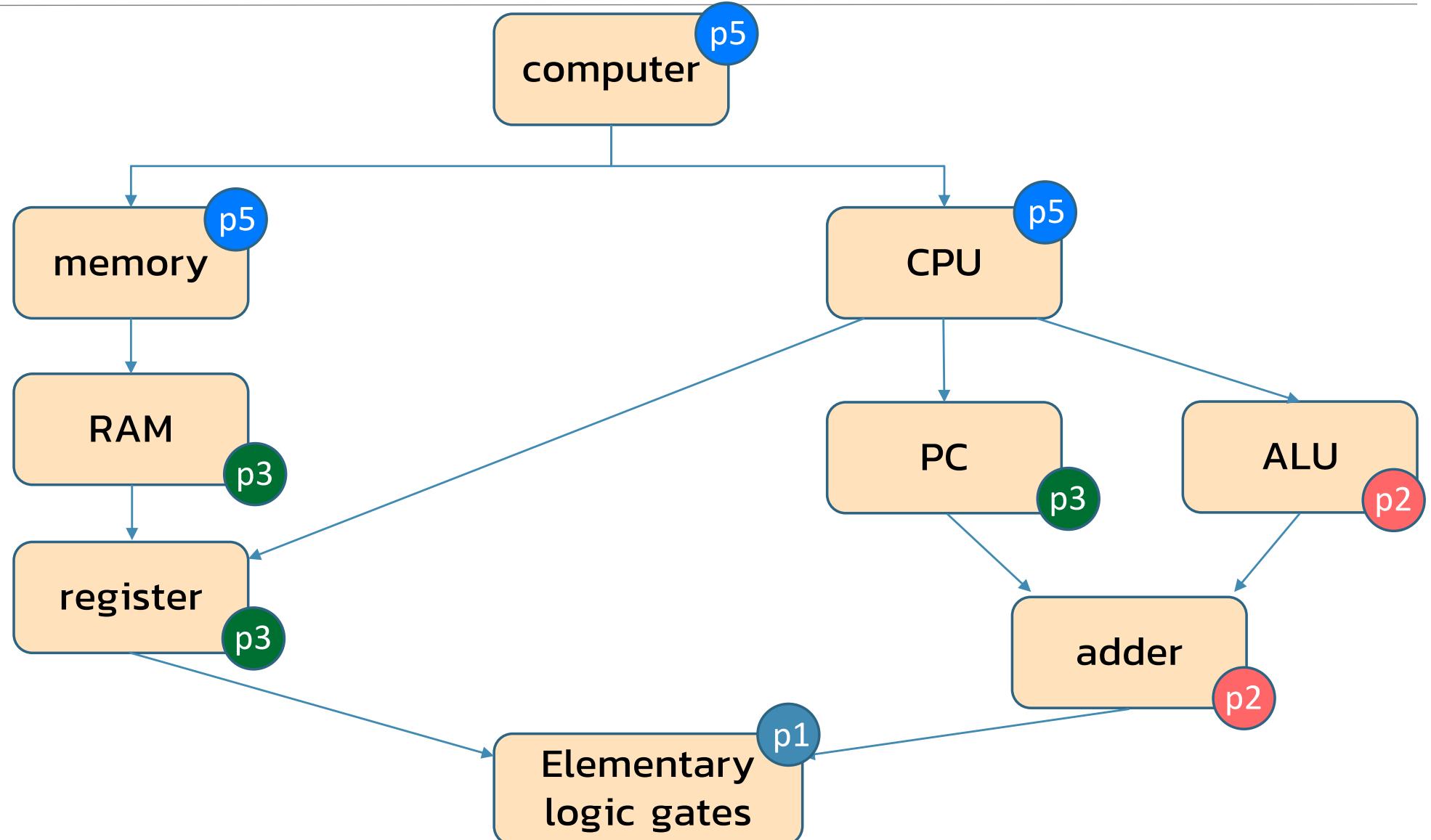




Hardware organization



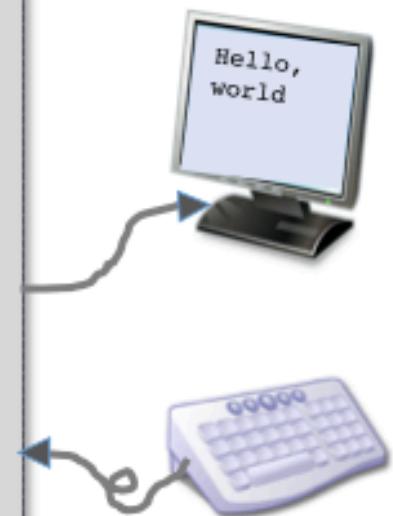
Hardware organization

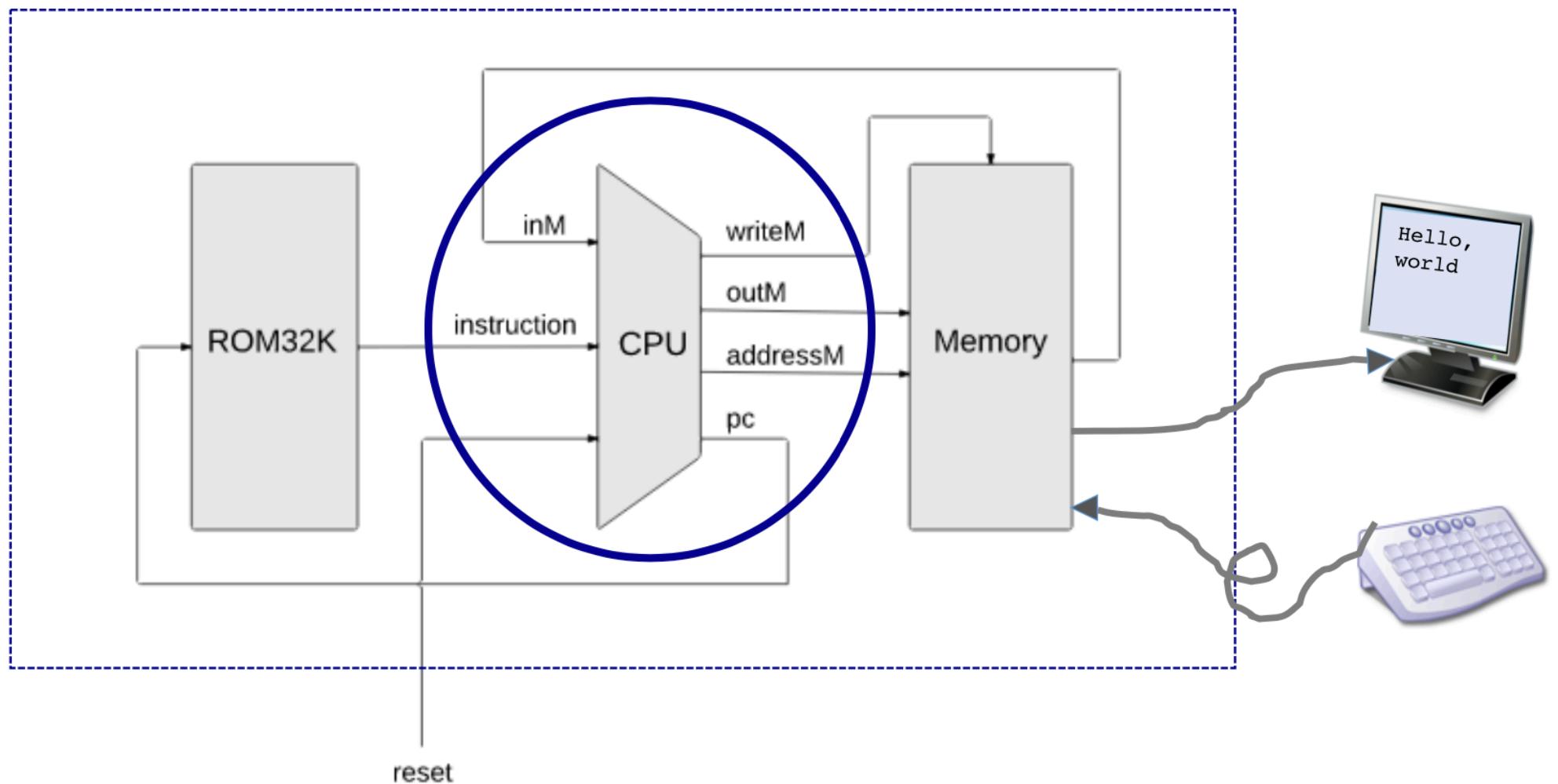


Hack Computer

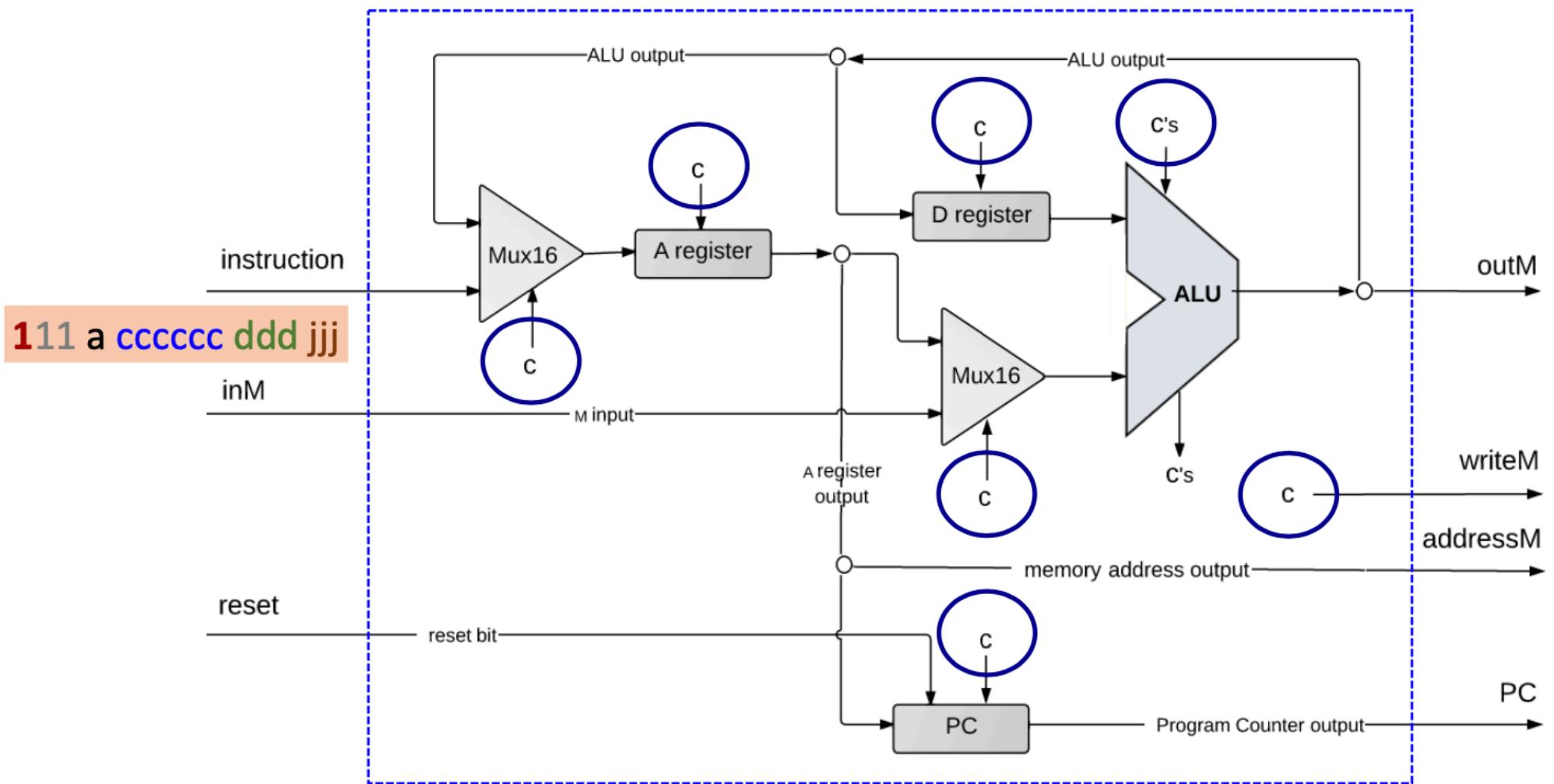


reset









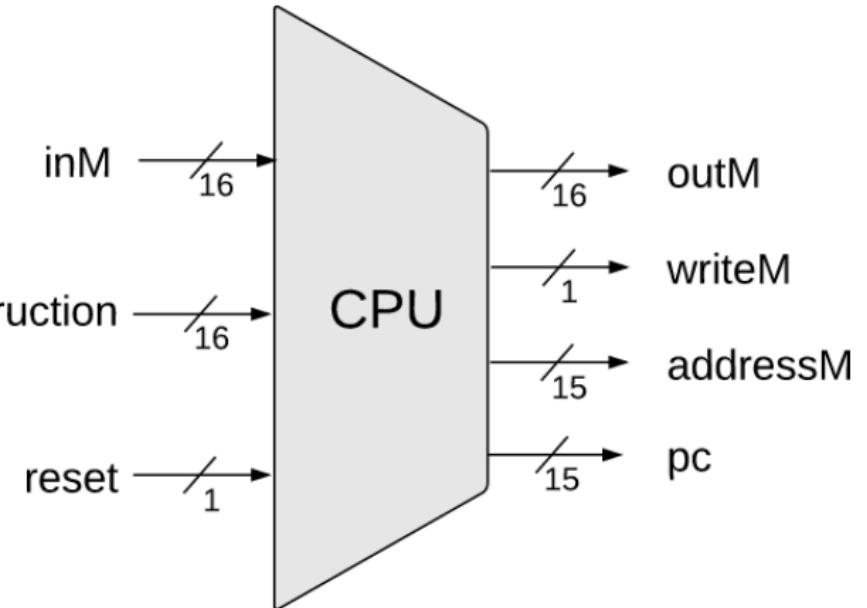
```

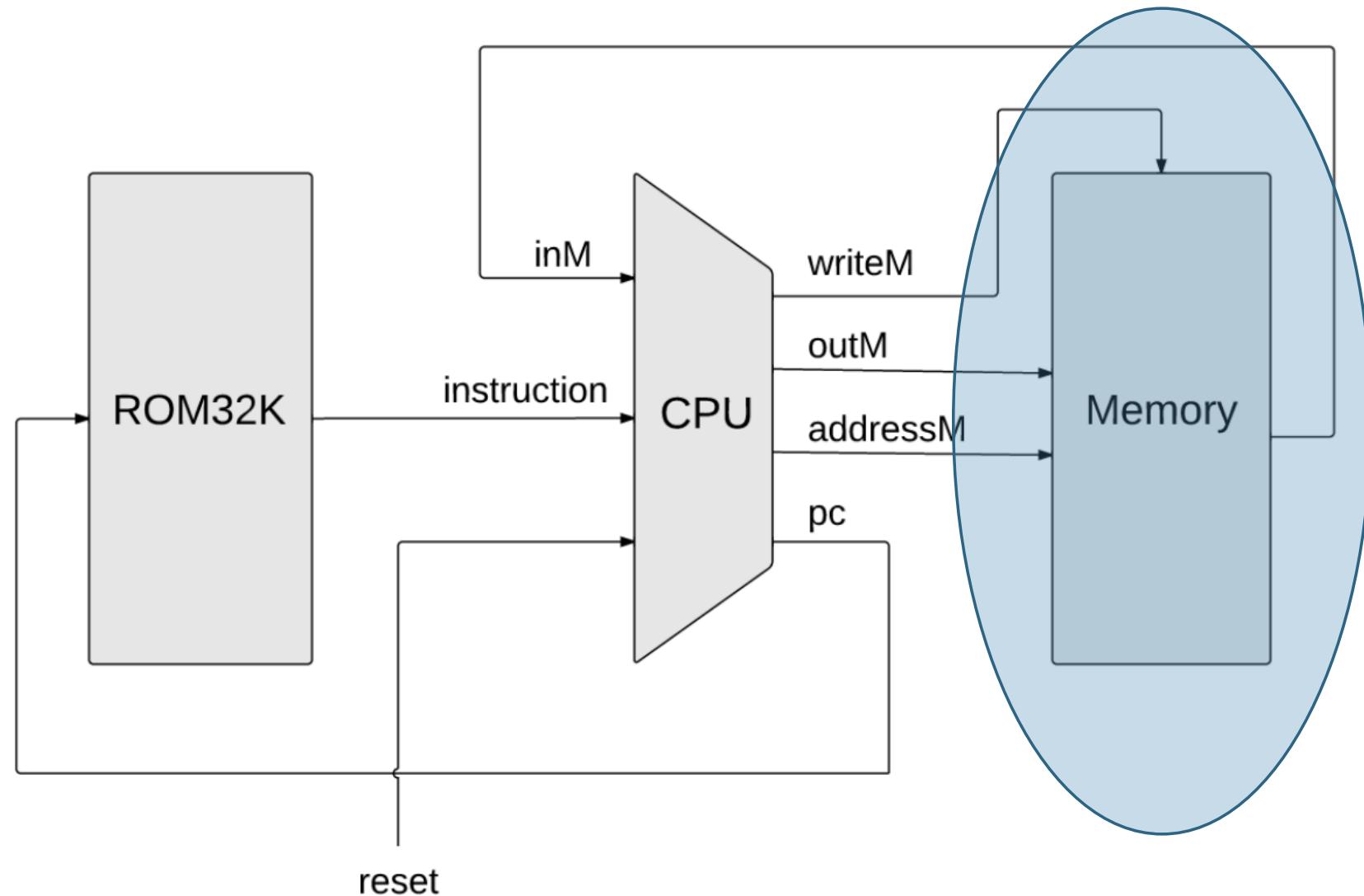
/**
 * The Central Processing unit (CPU)
 * Consists of an ALU and a set of
 registers, * designed to fetch and
 execute instructions * written in
 Hack machine language.
 */

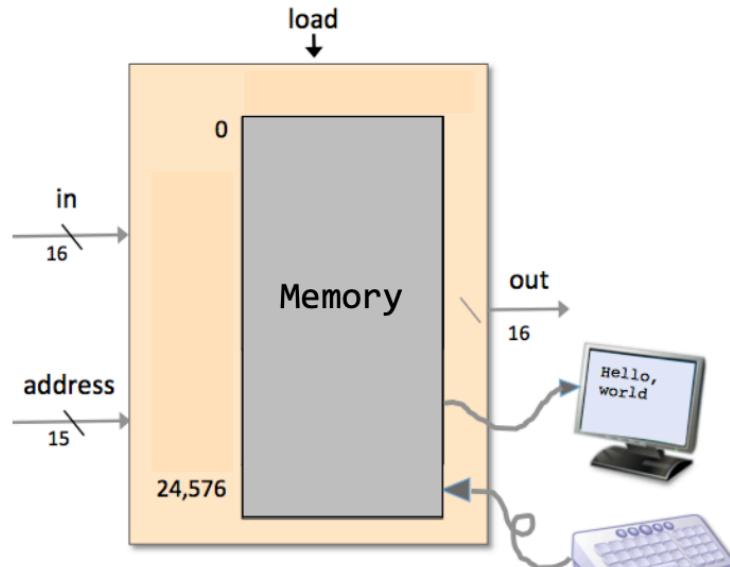
CHIP CPU {
    IN
        inM[16], // value of M = RAM[A]
        instruction[16], // for execution
        reset; // Signals for restart

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

```







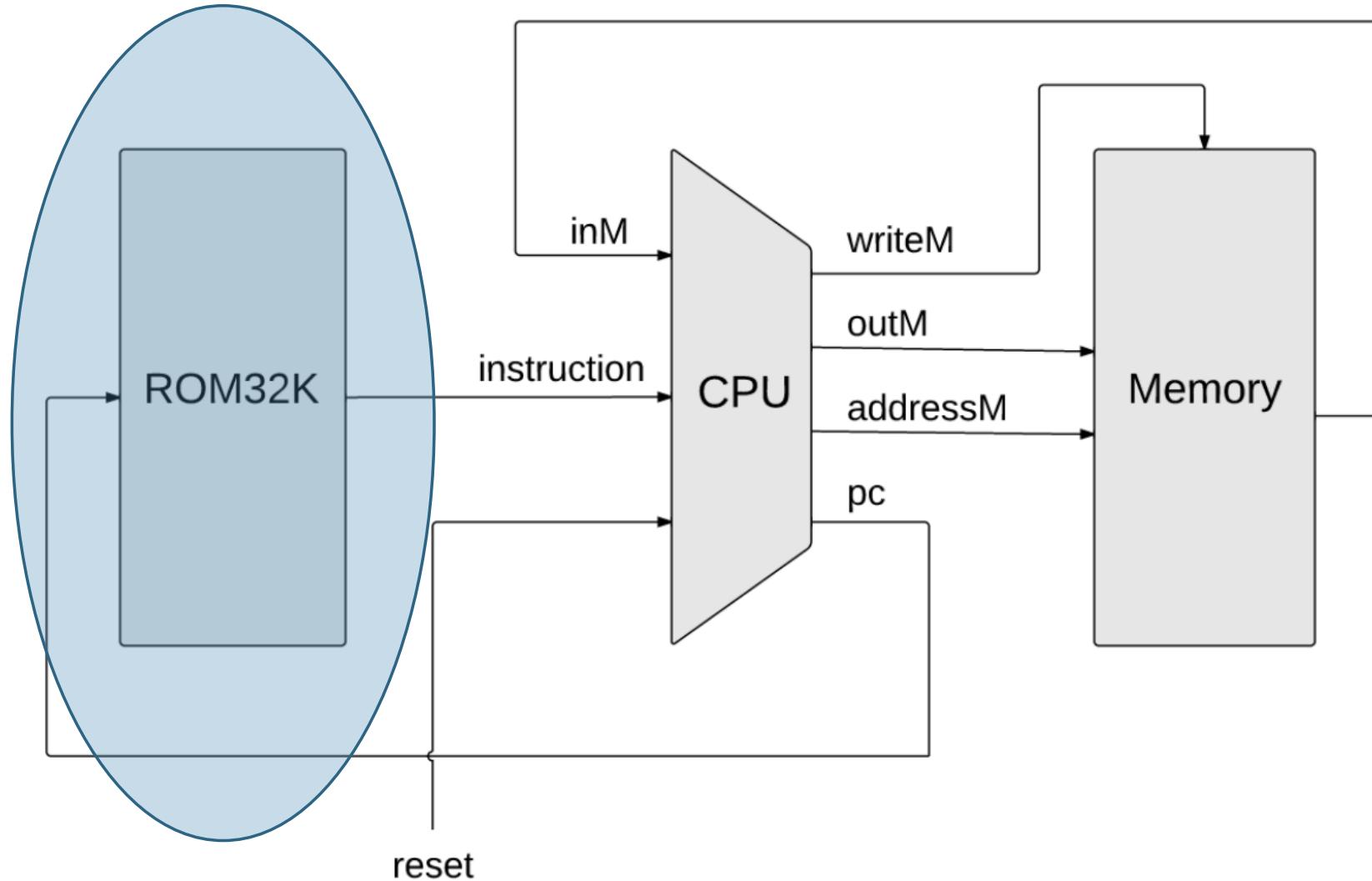
```
/** Memory of 16K 16-bit registers */
CHIP RAM16K {
    IN
    address[14],
    .
    .
    .
}

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

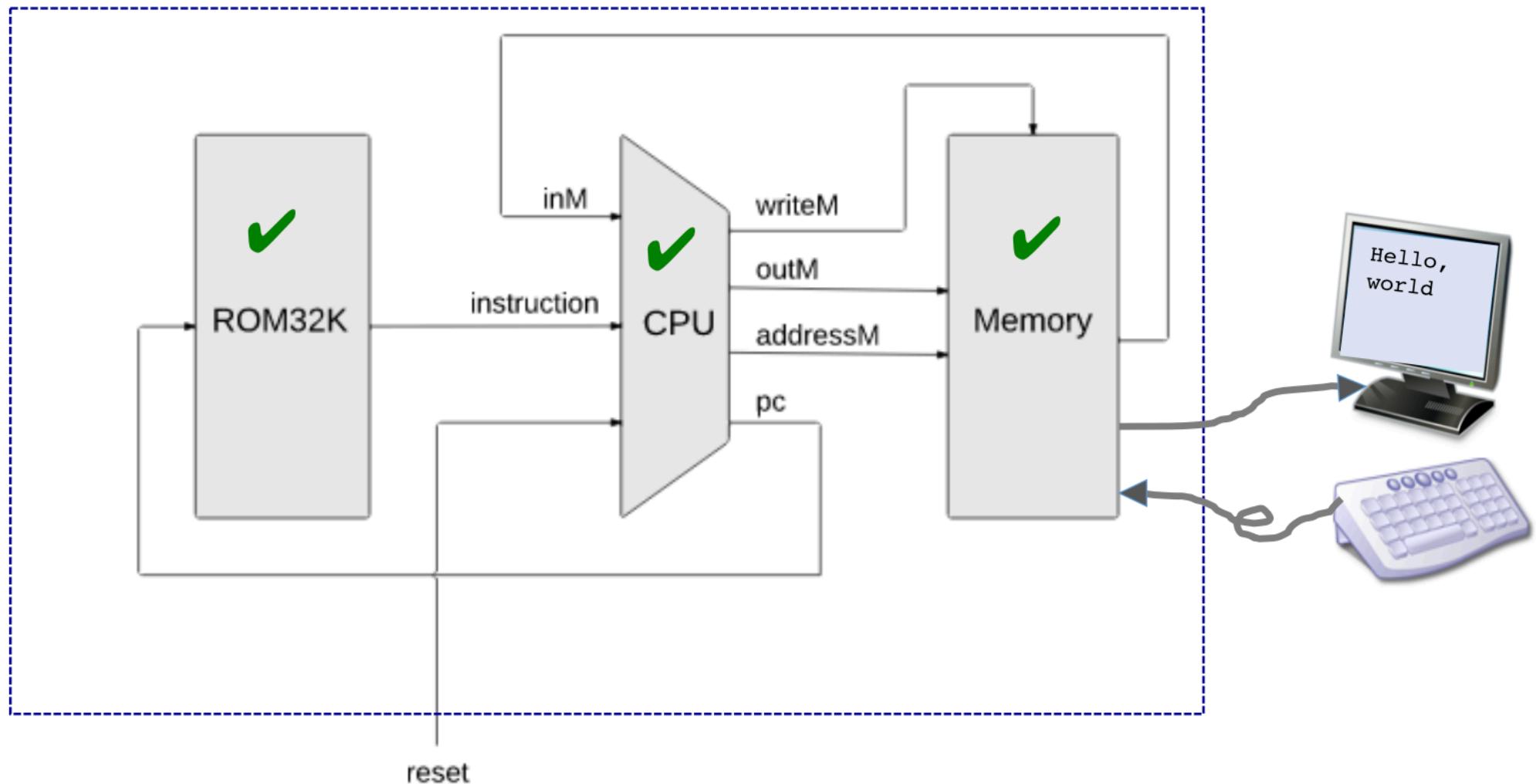
/** 16-bit register with a
 * keyboard input side effect */
CHIP Keyboard {
    OUT
    out[16];
    BUILTIN Keyboard;
}
```

built in project 3

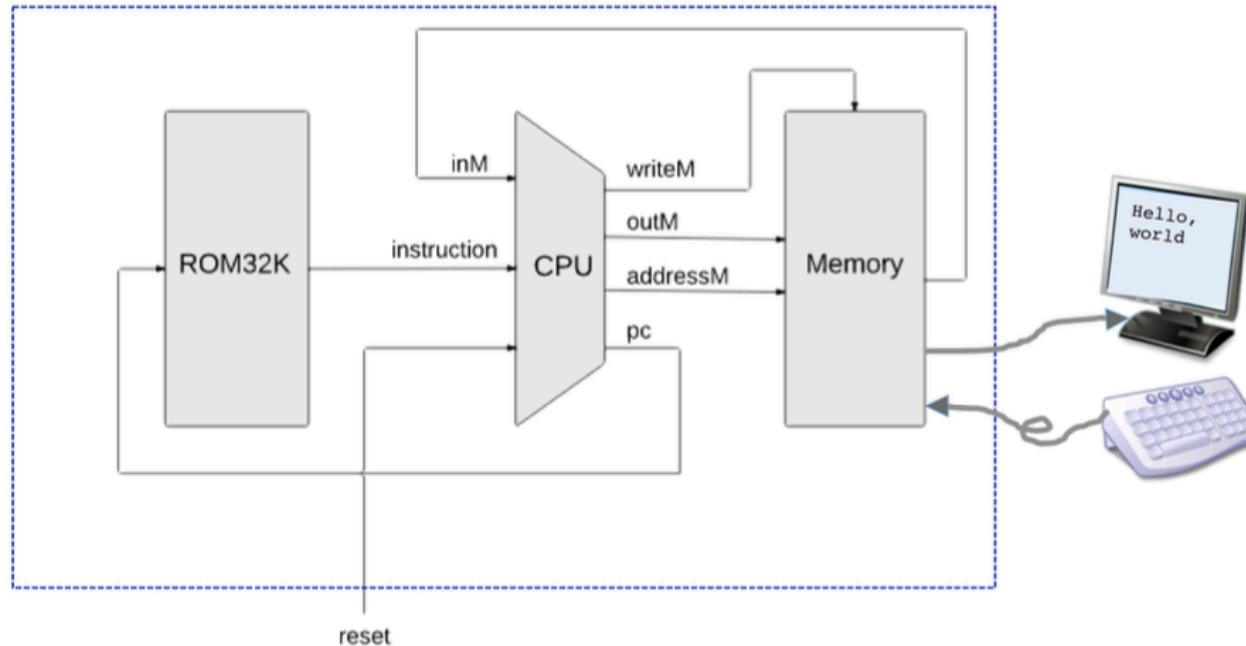
built-in chips



Hack Computer



Computer.hdl



```
CHIP Computer {  
    IN reset;  
  
    PARTS:  
        // your code here  
}
```

Coming up: W5.5 Project #5

โปรเจ็ค #5