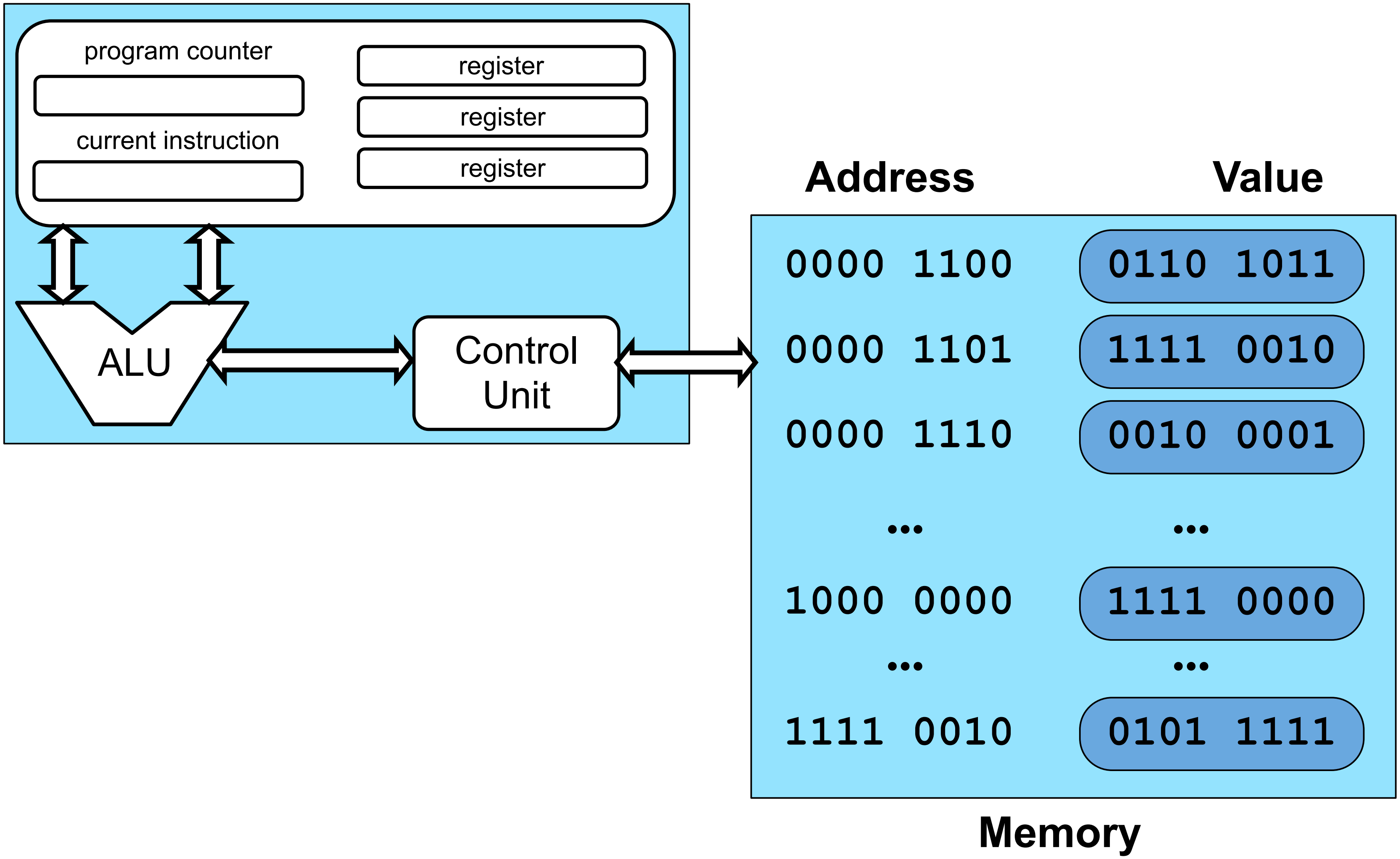
Computer Architecture and Programming: Examples and Sample Problems

ICS312

Machine-level and Systems Programming

Henri Casanova (henric@hawaii.edu)

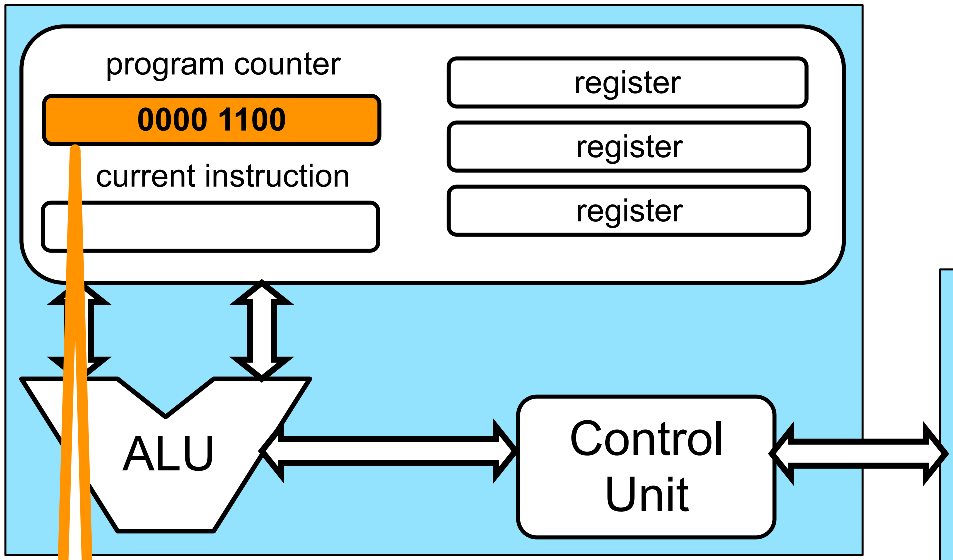
Fetch-Decode-Execute



Address	Value
0000 1100	0110 1011
0000 1101	1111 0010
0000 1110	0010 0001
…	…
1000 0000	1111 0000
…	…
1111 0010	0101 1111

Memory

Fetch-Decode-Execute

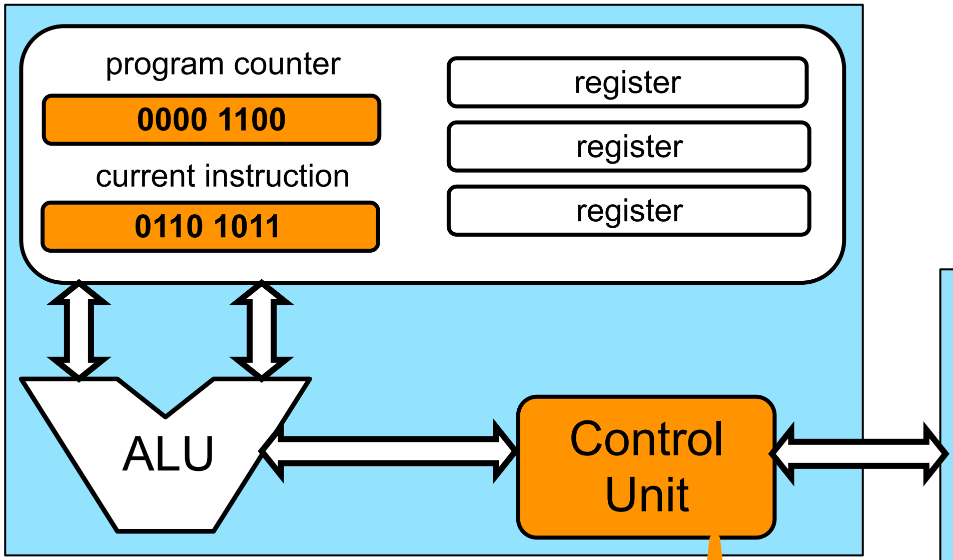


Address	Value
0000 1100	0110 1011
0000 1101	1111 0010
0000 1110	0010 0001
•••	•••
1000 0000	1111 0000
•••	•••
1111 0010	0101 1111

Memory

Somehow, the program counter is initialized to some content, which is an address (we’ll see how that happens much later)

Fetch-Decode-Execute

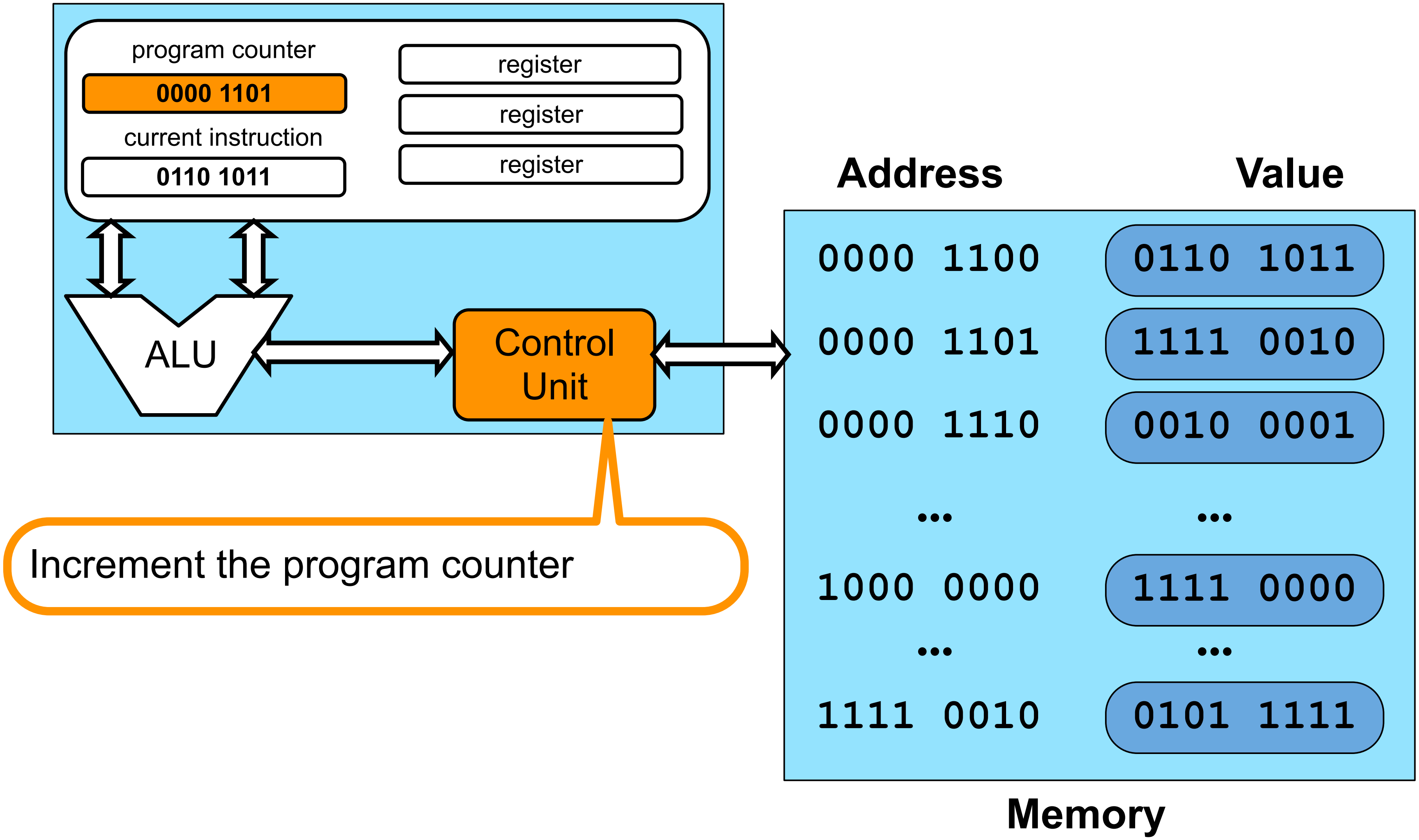


Address	Value
0000 1100	0110 1011
0000 1101	1111 0010
0000 1110	0010 0001
…	…
1000 0000	1111 0000
…	…
1111 0010	0101 1111

Memory

Fetch the content (instruction) at address 0000 1100, which is "0110 1011", and store it in the "current instruction" register

Fetch-Decode-Execute



program counter

0000 1101

current instruction

0110 1011

register

register

register

ALU

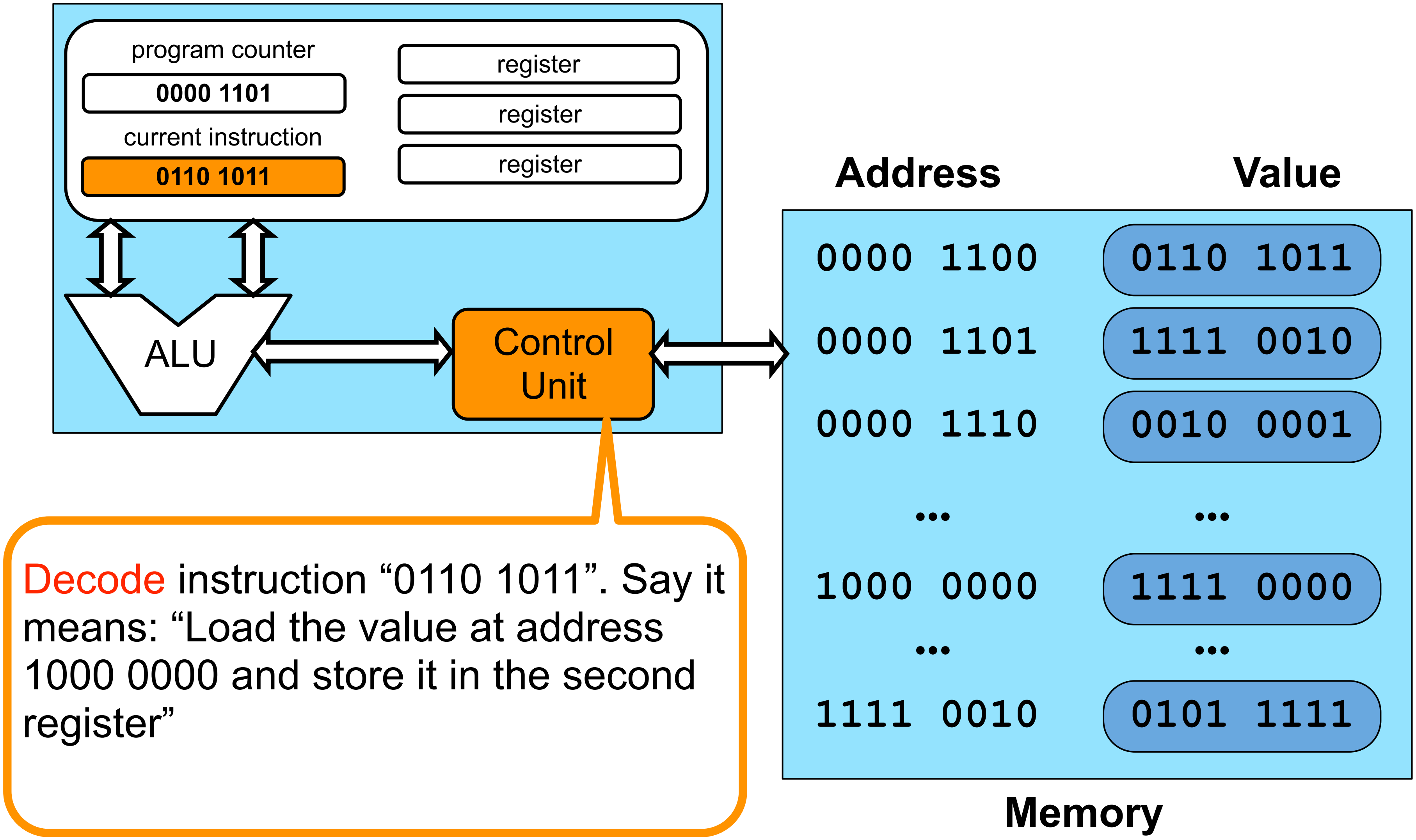
Control Unit

Address	Value
0000 1100	0110 1011
0000 1101	1111 0010
0000 1110	0010 0001
…	…
1000 0000	1111 0000
…	…
1111 0010	0101 1111

Memory

Increment the program counter

Fetch-Decode-Execute

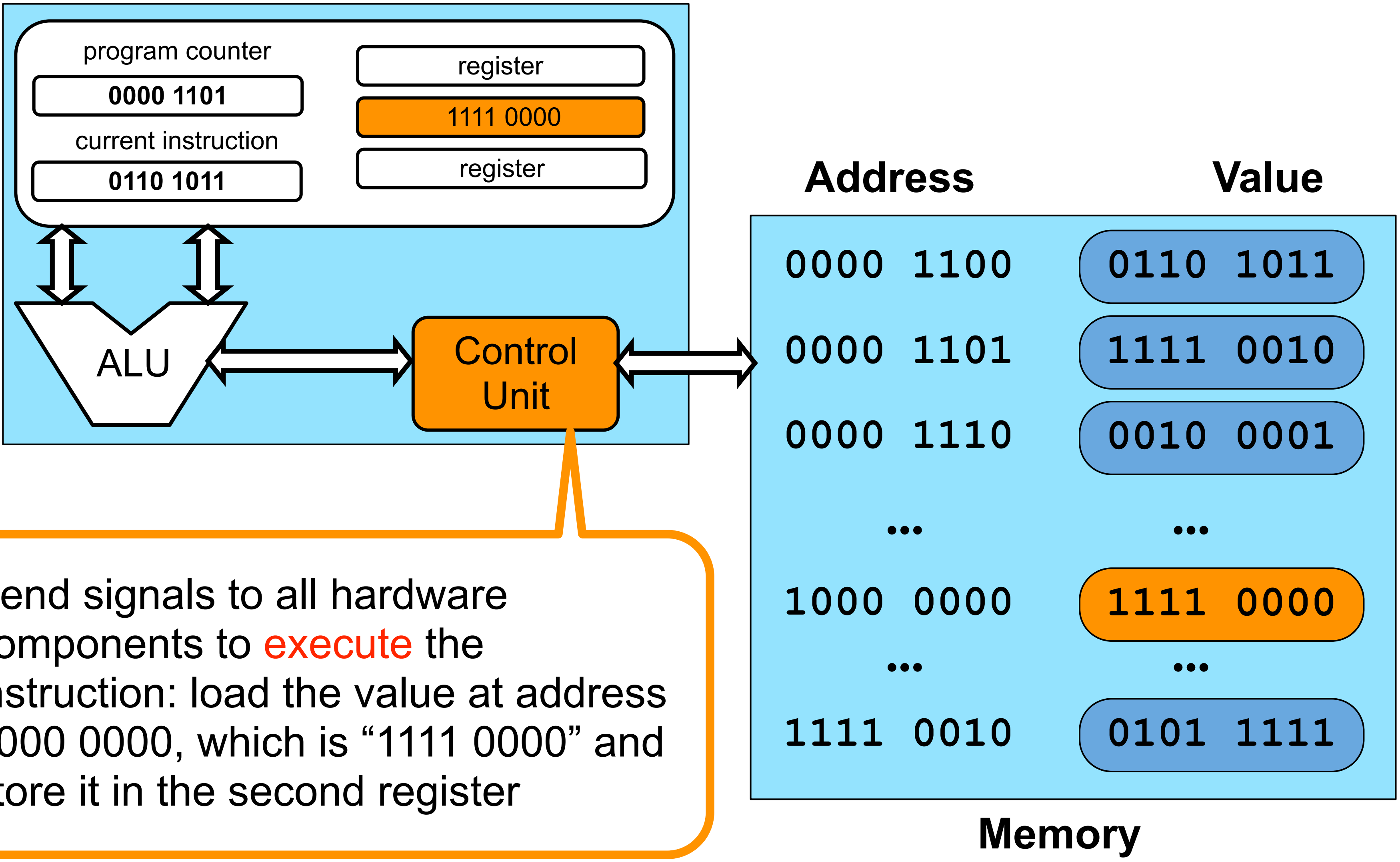


Address	Value
0000 1100	0110 1011
0000 1101	1111 0010
0000 1110	0010 0001
…	…
1000 0000	1111 0000
…	…
1111 0010	0101 1111

Memory

Decode instruction “0110 1011”. Say it means: “Load the value at address 1000 0000 and store it in the second register”

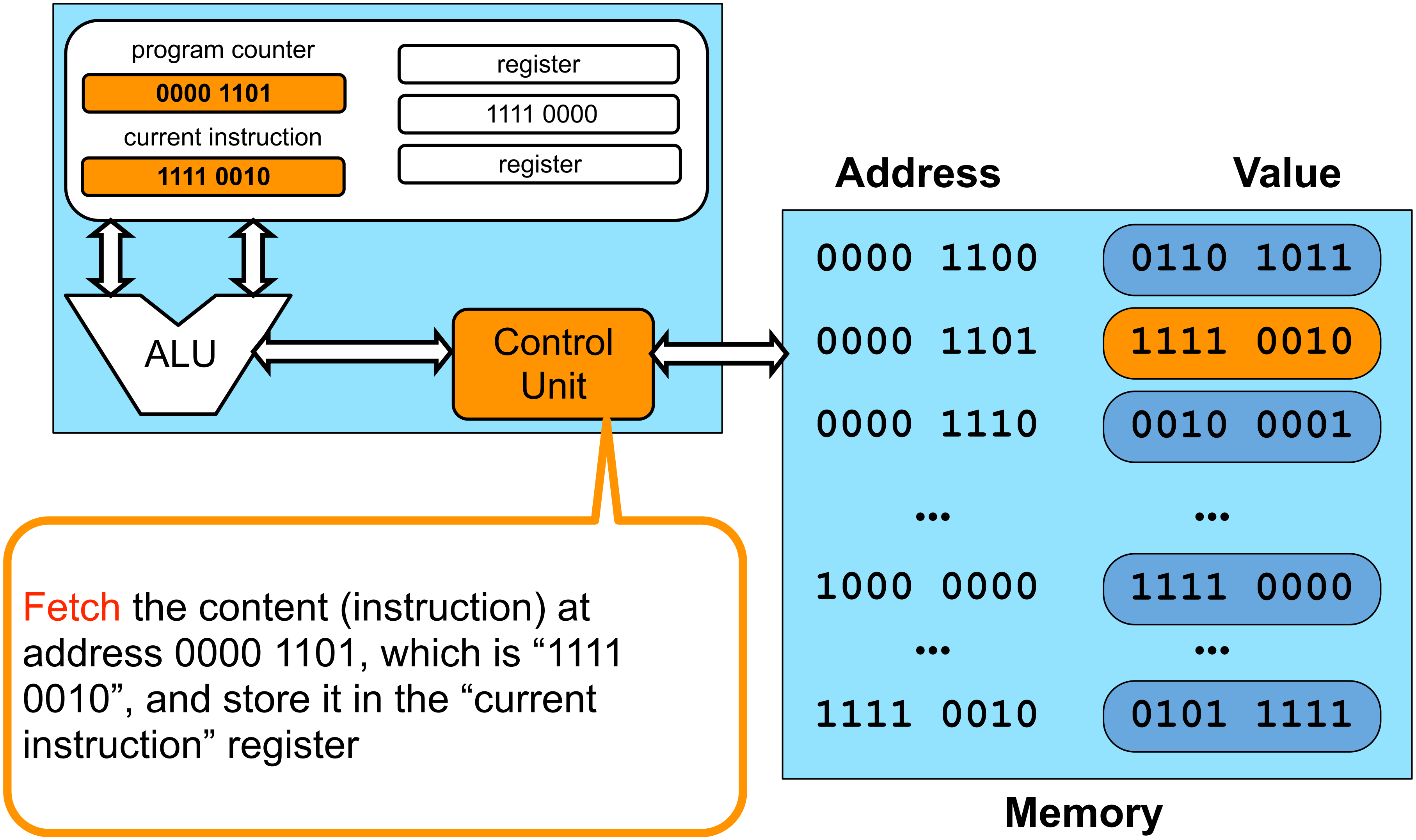
Fetch-Decode-Execute



Address	Value
0000 1100	0110 1011
0000 1101	1111 0010
0000 1110	0010 0001
…	…
1000 0000	1111 0000
…	…
1111 0010	0101 1111

Send signals to all hardware components to execute the instruction: load the value at address 1000 0000, which is “1111 0000” and store it in the second register

Fetch-Decode-Execute

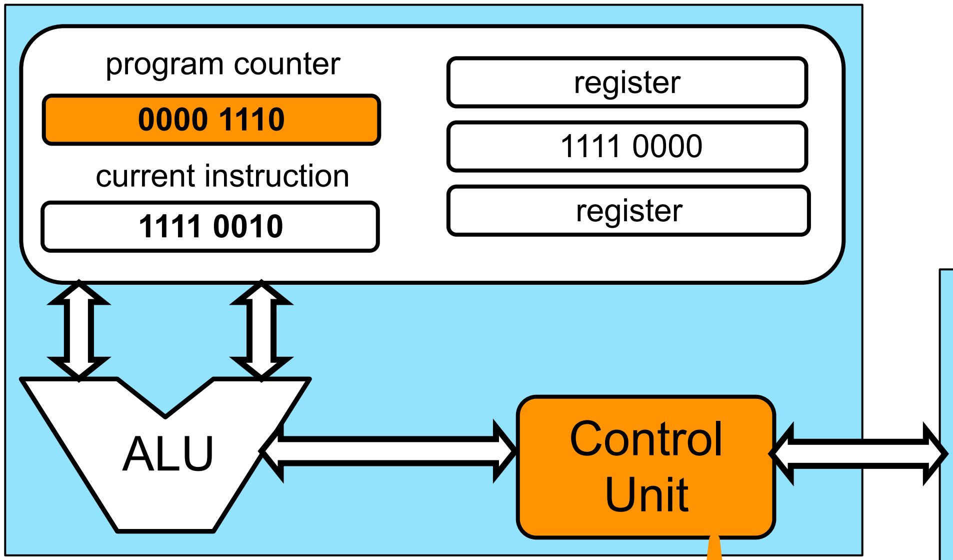


Fetch the content (instruction) at address 0000 1101, which is “1111 0010”, and store it in the “current instruction” register

Address	Value
0000 1100	0110 1011
0000 1101	1111 0010
0000 1110	0010 0001
…	…
1000 0000	1111 0000
…	…
1111 0010	0101 1111

Memory

Fetch-Decode-Execute

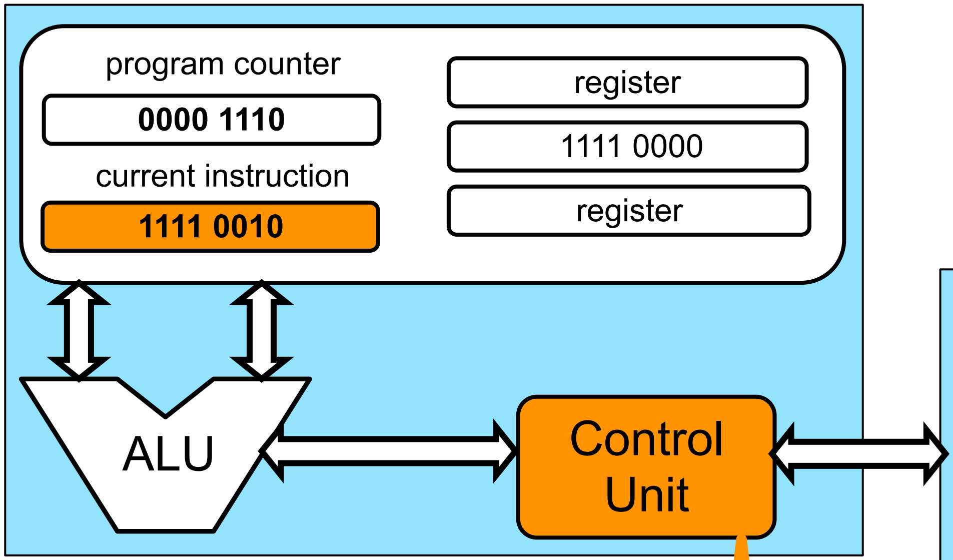


Increment the program counter

Address	Value
0000 1100	0110 1011
0000 1101	1111 0010
0000 1110	0010 0001
…	…
1000 0000	1111 0000
…	…
1111 0010	0101 1111

Memory

Fetch-Decode-Execute

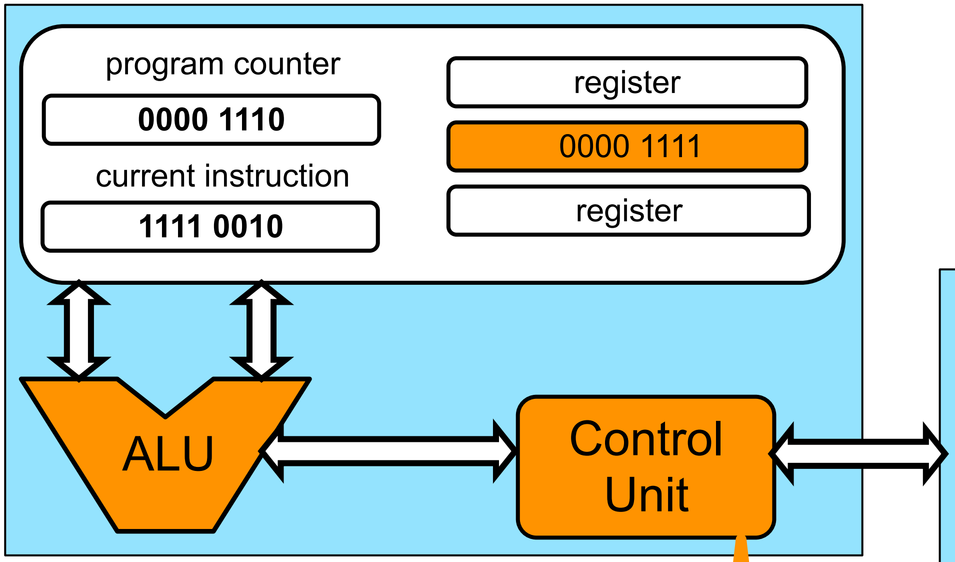


Address	Value
0000 1100	0110 1011
0000 1101	1111 0010
0000 1110	0010 0001
...	...
1000 0000	1111 0000
...	...
1111 0010	0101 1111

Memory

Decode instruction “1111 0010”. Say it means: “Do a logical NOT on the second register”

Fetch-Decode-Execute



program counter

0000 1110

current instruction

1111 0010

register

0000 1111

register

ALU

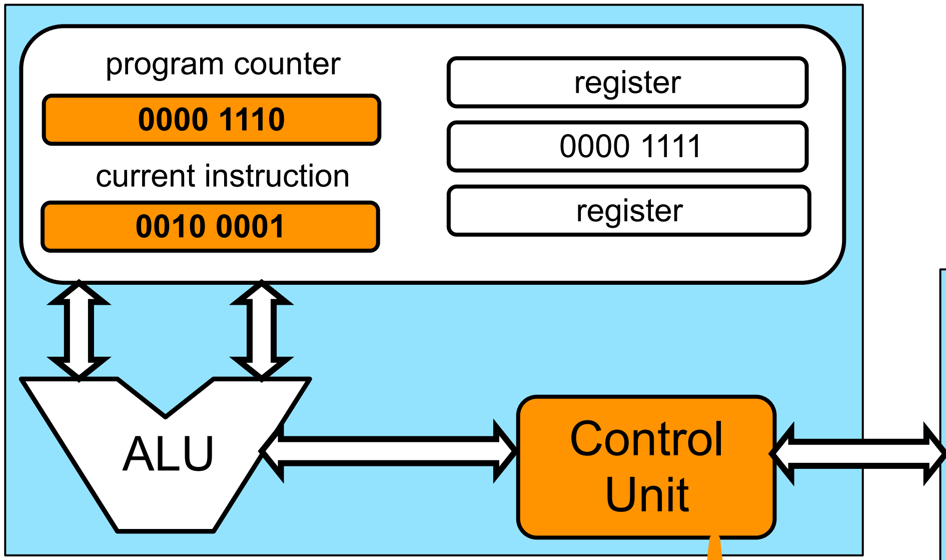
Control Unit

Address	Value
0000 1100	0110 1011
0000 1101	1111 0010
0000 1110	0010 0001
…	…
1000 0000	1111 0000
…	…
1111 0010	0101 1111

Memory

Send signals to all hardware components to execute the instruction: do a logical NOT on the second register

Fetch-Decode-Execute



program counter

0000 1110

current instruction

0010 0001

register

0000 1111

register

ALU

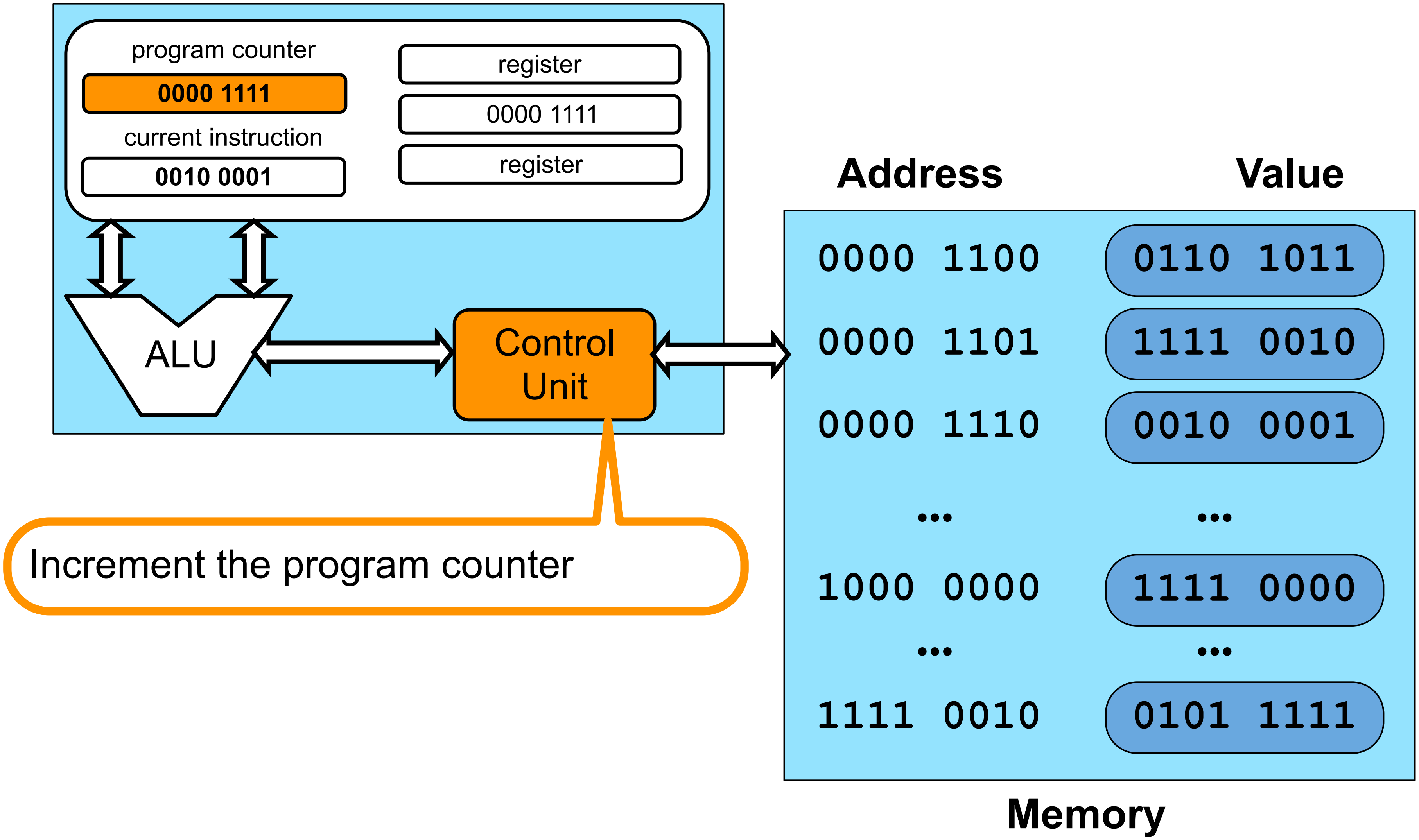
Control Unit

Address	Value
0000 1100	0110 1011
0000 1101	1111 0010
0000 1110	0010 0001
…	…
1000 0000	1111 0000
…	…
1111 0010	0101 1111

Memory

Fetch the content (instruction) at address 0000 1110, which is “0010 0001”, and store it in the “current instruction” register

Fetch-Decode-Execute

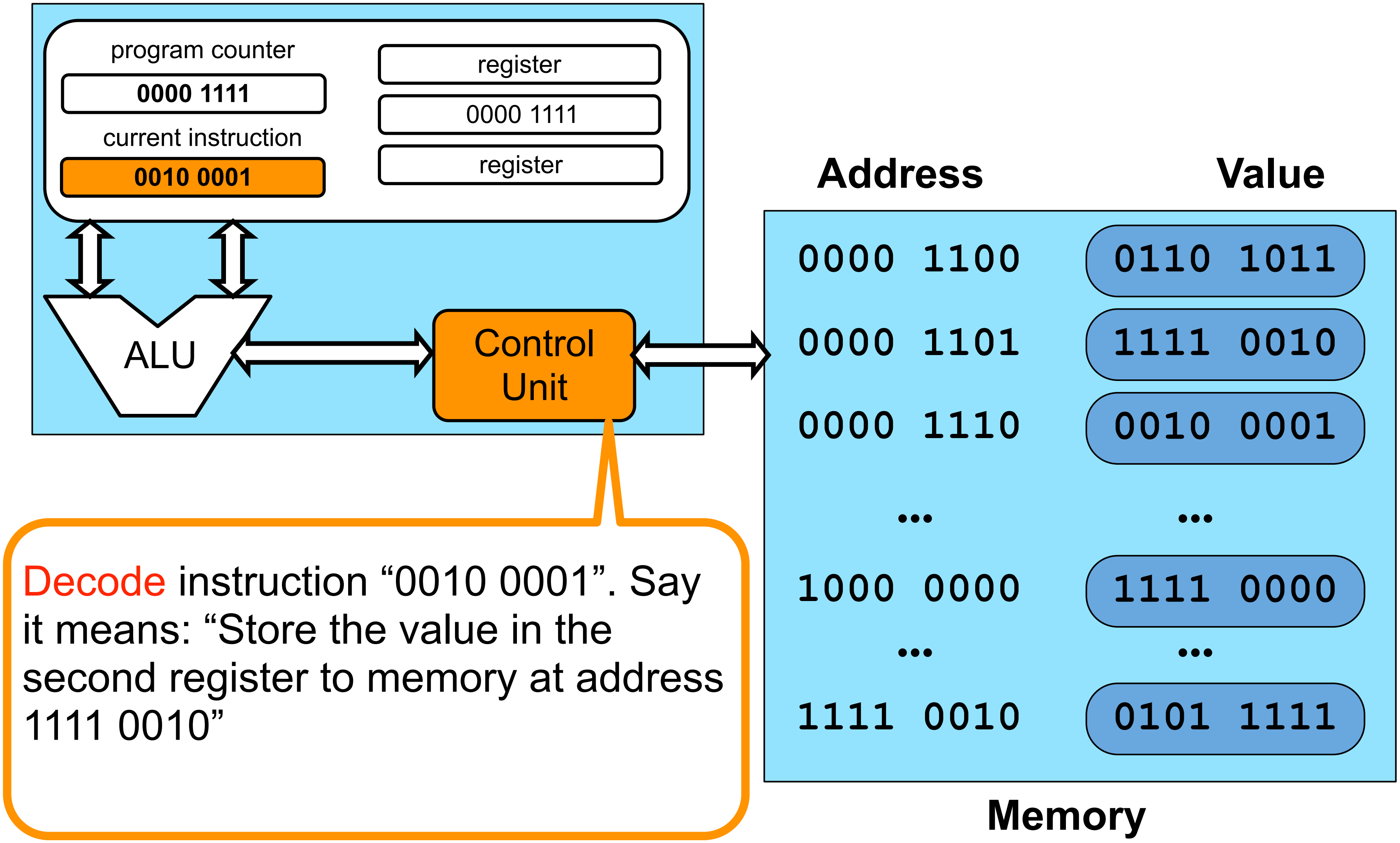


Increment the program counter

Address	Value
0000 1100	0110 1011
0000 1101	1111 0010
0000 1110	0010 0001
…	…
1000 0000	1111 0000
…	…
1111 0010	0101 1111

Memory

Fetch-Decode-Execute

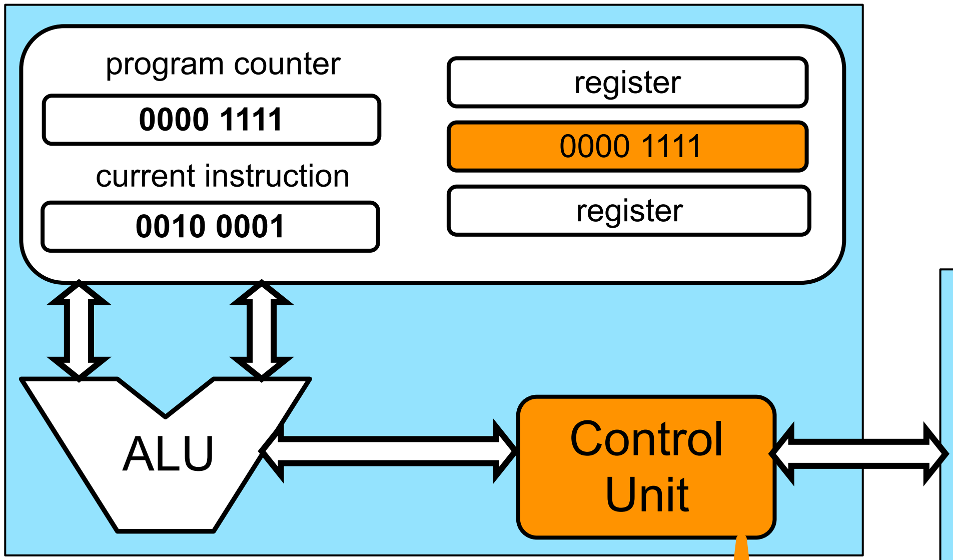


Decode instruction “0010 0001”. Say it means: “Store the value in the second register to memory at address 1111 0010”

Address	Value
0000 1100	0110 1011
0000 1101	1111 0010
0000 1110	0010 0001
…	…
1000 0000	1111 0000
…	…
1111 0010	0101 1111

Memory

Fetch-Decode-Execute



Send signals to all hardware components to execute the instruction: store the value in the second register, which is 0000 1111, to memory at address 1111 0010

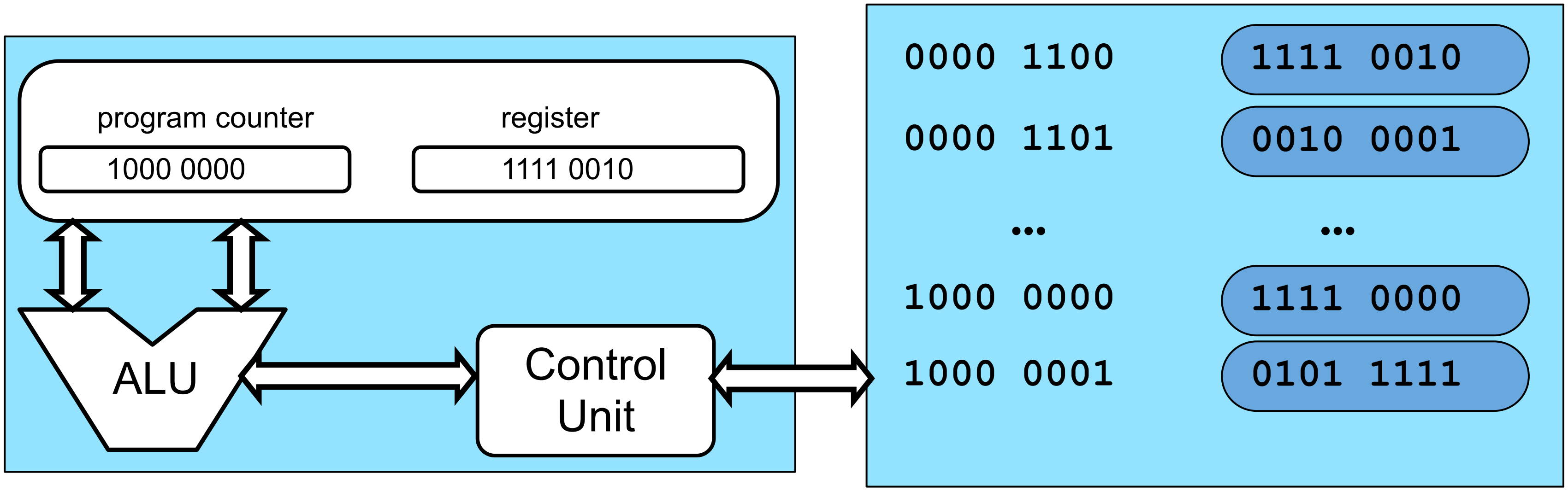
Address	Value
0000 1100	0110 1011
0000 1101	1111 0010
0000 1110	0010 0001
…	…
1000 0000	1111 0000
…	…
1111 0010	0000 1111

Memory

Practice

- With the following instruction set definition and machine state, what is the new memory state after execution completes?

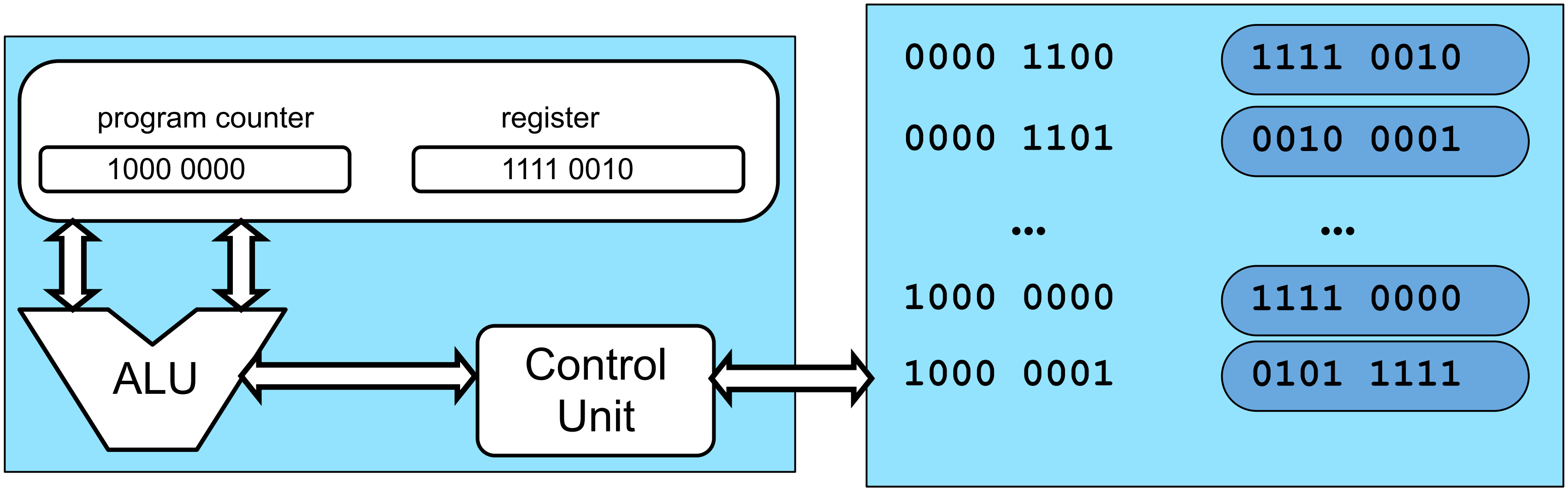
code	operation
1111 0000	Increment the register
1111 0010	Decrement the register
0101 1111	Save register to address not(register)



0000 1100	1111 0010
0000 1101	0010 0001
…	…
1000 0000	1111 0000
1000 0001	0101 1111

- Fetch the instruction: “1111 0000”

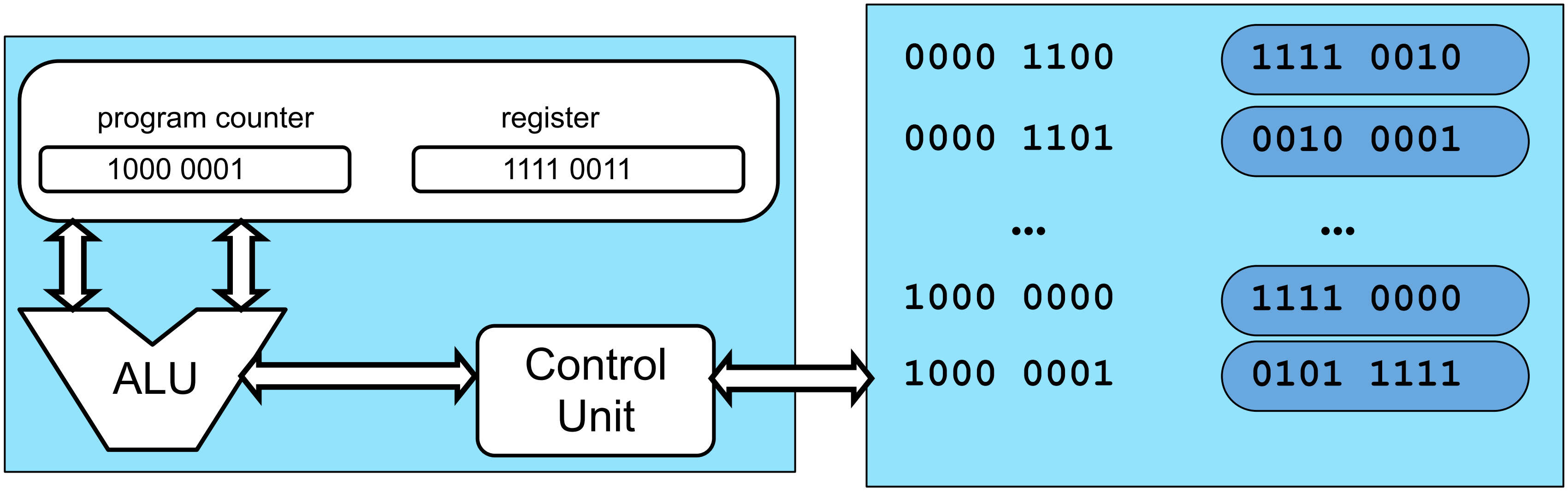
code	operation
1111 0000	Increment the register
1111 0010	Decrement the register
0101 1111	Save register to address not(register)



0000 1100	1111 0010
0000 1101	0010 0001
…	…
1000 0000	1111 0000
1000 0001	0101 1111

- Fetch the instruction: “1111 0000”
- Execute it: increment register to value “1111 0011”

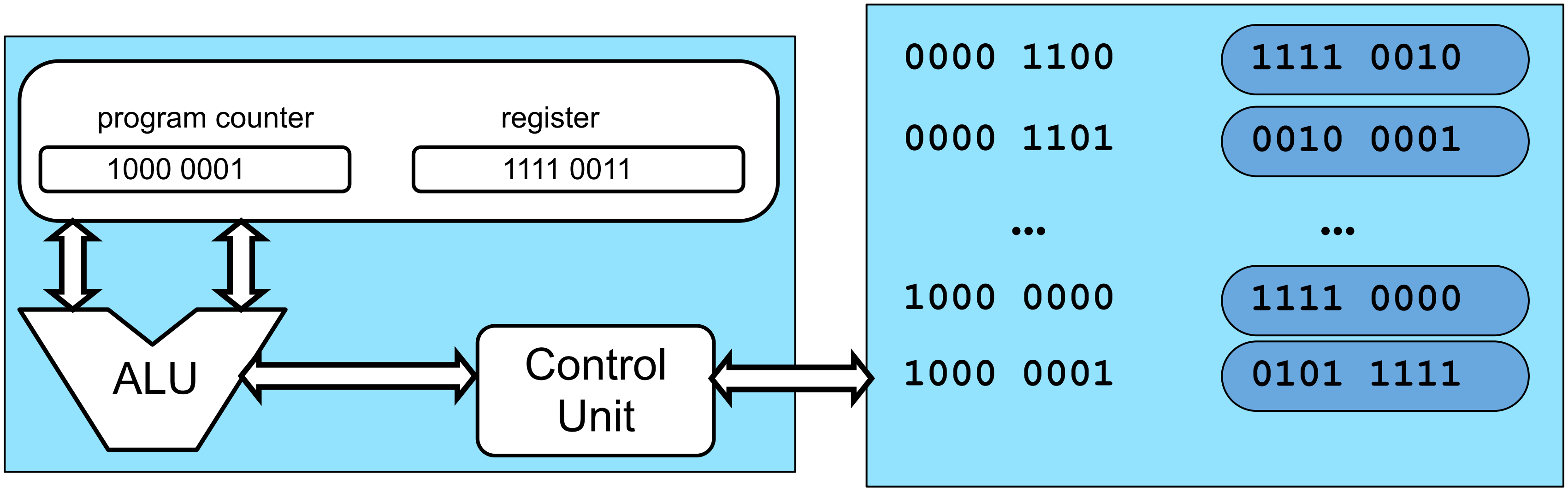
code	operation
1111 0000	Increment the register
1111 0010	Decrement the register
0101 1111	Save register to address not(register)



0000 1100	1111 0010
0000 1101	0010 0001
...	...
1000 0000	1111 0000
1000 0001	0101 1111

- Fetch the instruction: “1111 0000”
- Execute it: increment register to value “1111 0011”
- Fetch the next instruction: “0101 1111”
-

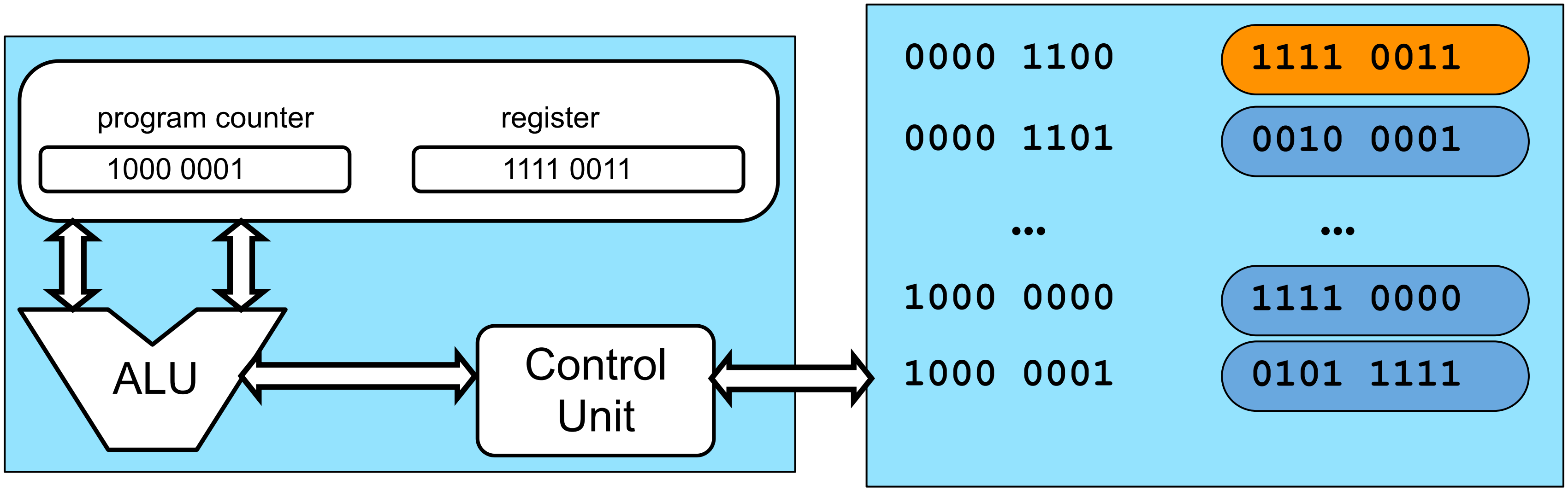
code	operation
1111 0000	Increment the register
1111 0010	Decrement the register
0101 1111	Save register to address not(register)



0000 1100	1111 0010
0000 1101	0010 0001
…	…
1000 0000	1111 0000
1000 0001	0101 1111

- Fetch the instruction: “1111 0000”
- Execute it: increment register to value “1111 0011”
- Fetch the next instruction: “0101 1111”
- Execute it: save value “1111 0011” to address “0000 1100”

code	operation
1111 0000	Increment the register
1111 0010	Decrement the register
0101 1111	Save register to address not(register)



program counter: 1000 0001

register: 1111 0011

0000 1100	1111 0011
0000 1101	0010 0001
…	…
1000 0000	1111 0000
1000 0001	0101 1111