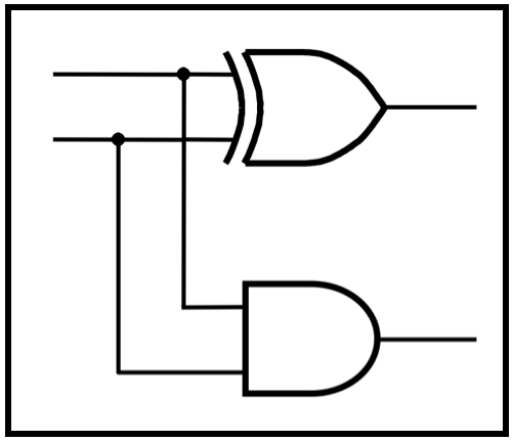


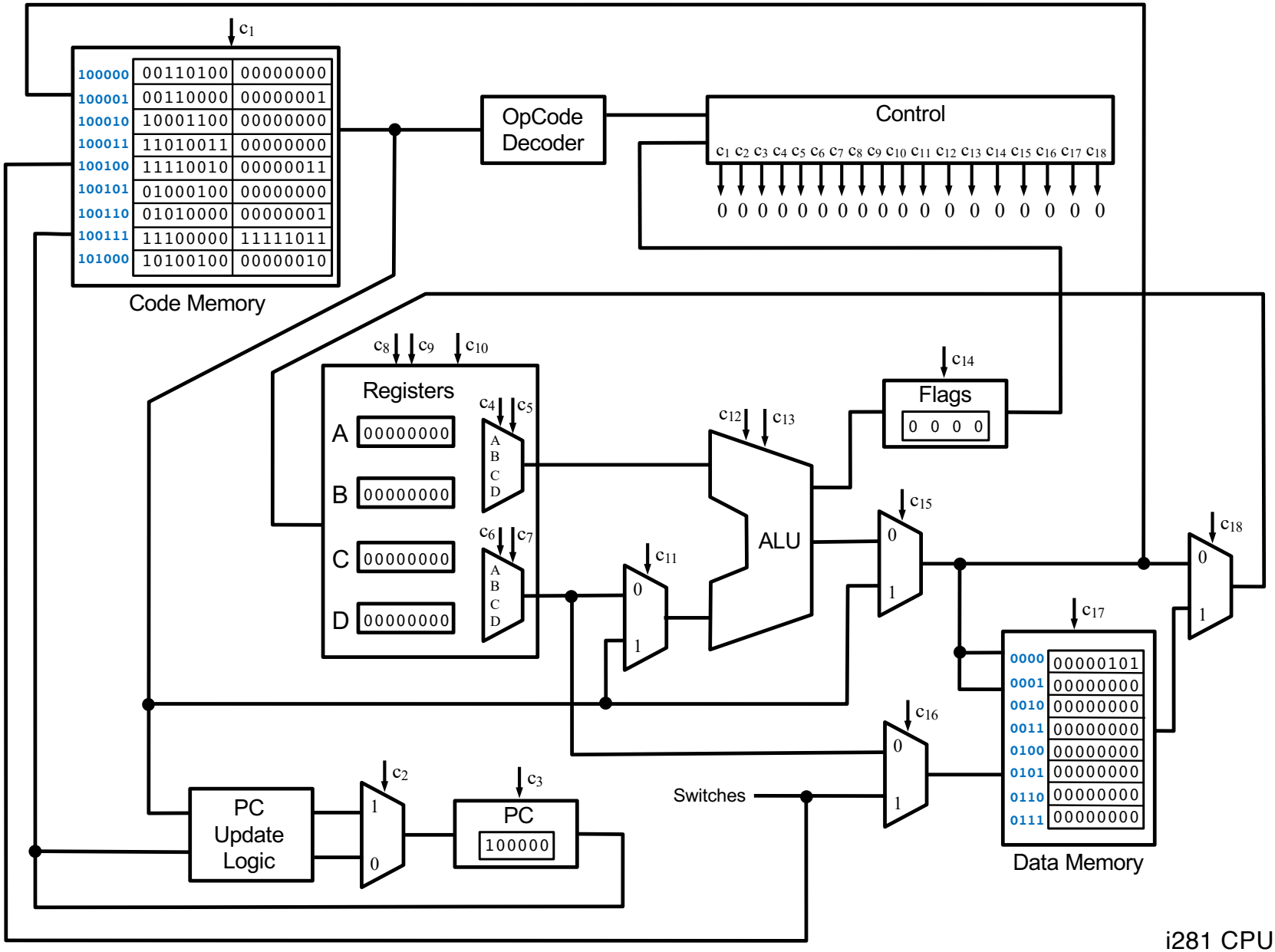
CprE 2810: Digital Logic

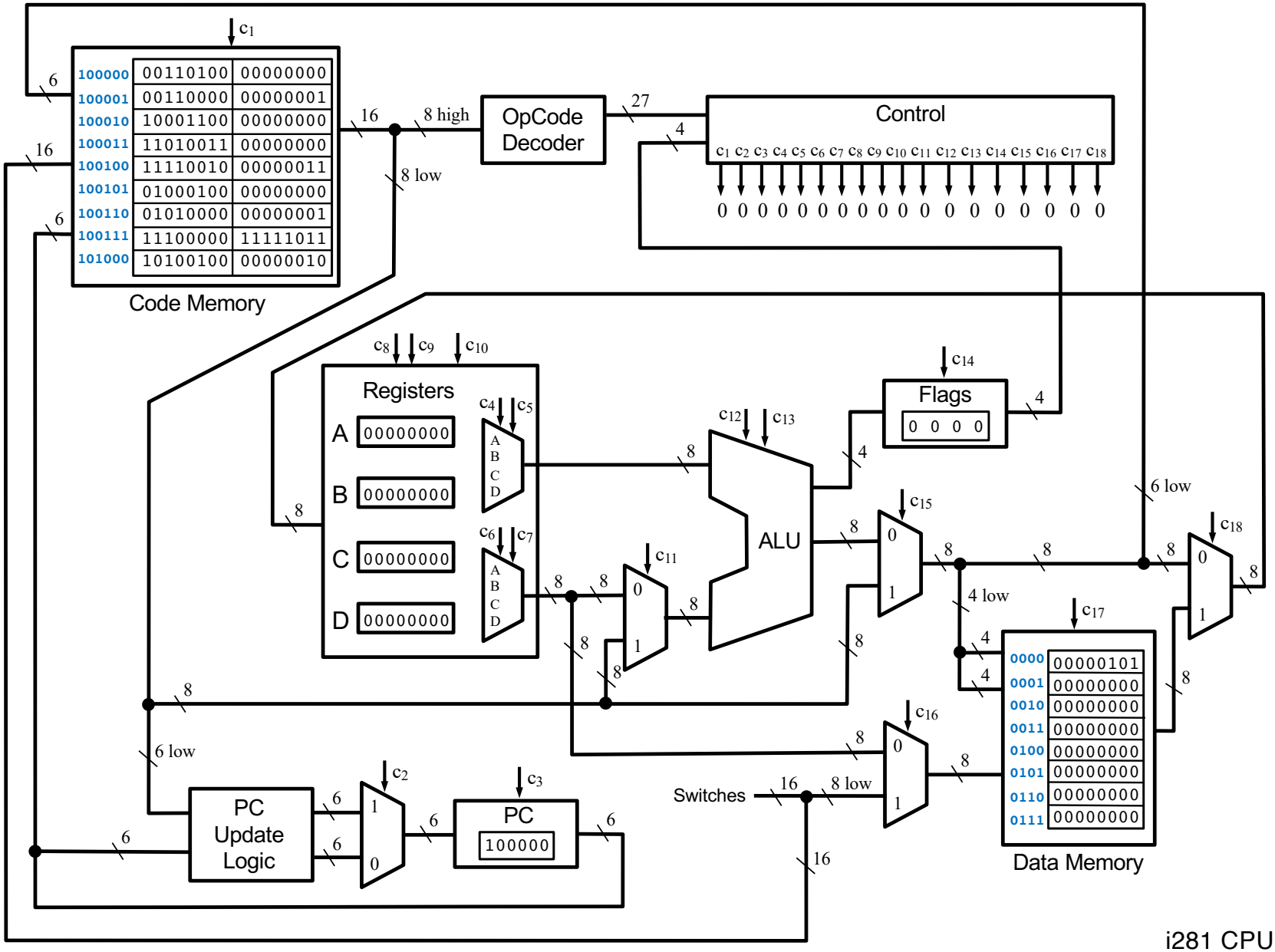
Instructor: Alexander Stoytchev

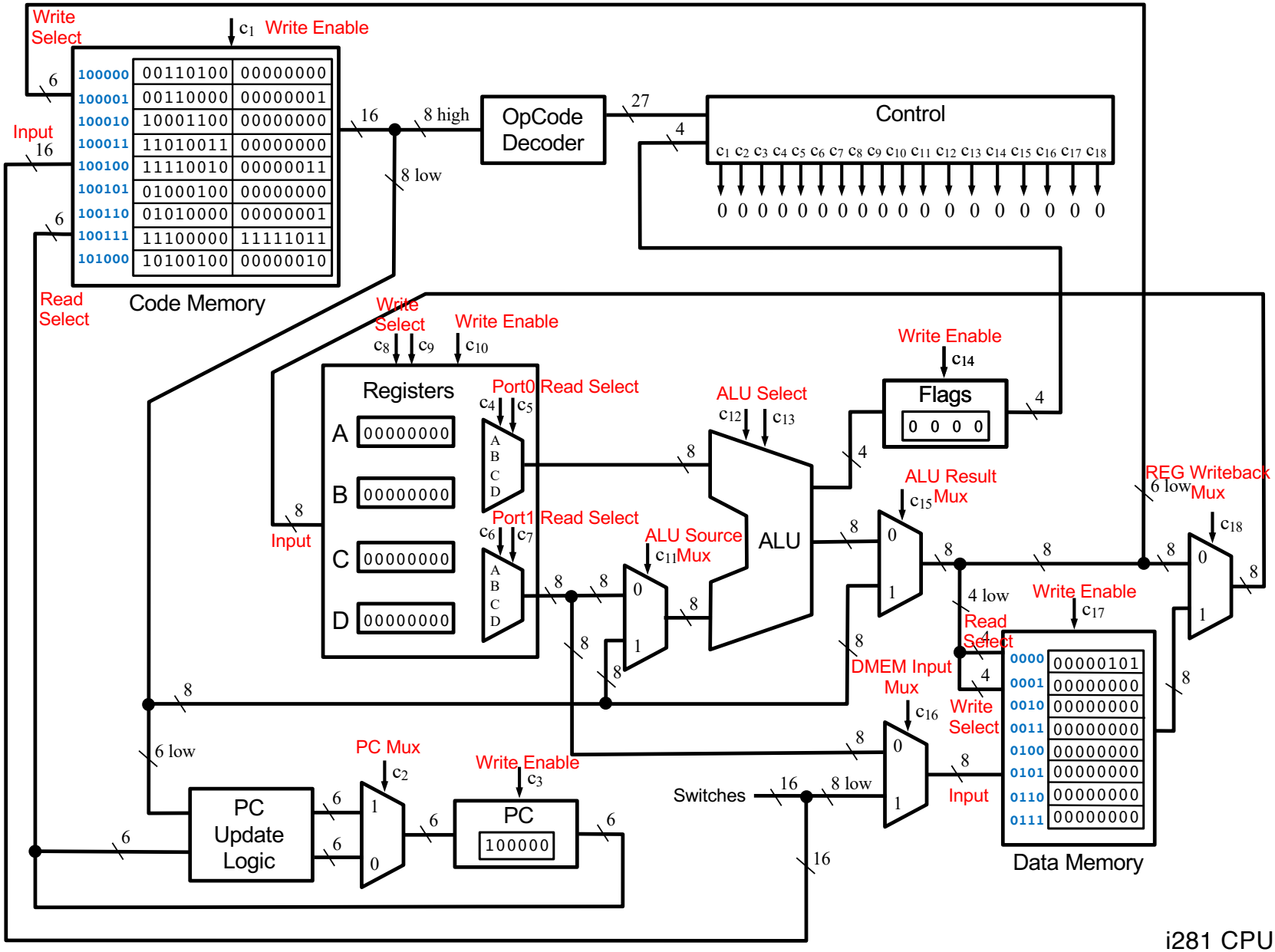
http://www.ece.iastate.edu/~alexs/classes/

i281 CPU Architecture

CprE 2810: Digital Logic
Iowa State University, Ames, IA


c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
OpCode Decoder
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
c11
c12 c13
ALU
c14
Flags
0 0 0 0
c15
c16
c17
c18
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
Switches
c2
c3
PC Update Logic
PC
100000
i281 CPU


c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c14
Flags
0 0 0 0
4
c12 c13
ALU
c15
c11
c16
c17
c18
6 low
4 low
8 low
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
Switches
16
c2
c3
PC Update Logic
PC
100000
i281 CPU


Write Select
c1 Write Enable
6
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
Input
16
6
Read Select
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Write Select
c8 c9
Write Enable
c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
Port0 Read Select
c4 c5
Port1 Read Select
c6 c7
A B C D
8
Input
ALU Select
c12 c13
ALU
ALU Source
c11 Mux
Write Enable
c14
Flags
0 0 0 0
4
ALU Result
c15 Mux
REG Writeback
6 low Mux
c18
4 low
Read Select
Write Enable
c17
DMEM Input
Mux
c16
Write Select
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
6 low
PC Mux
c2
Write Enable
c3
PC Update Logic
PC
100000
Switches
16
8 low
Input
i281 CPU

Drawing Convention

- Inputs enter from the left (for each box)
- Outputs exit from the right (for each box)
- Control lines are vertical arrows (come from the top)

i281 CPU

- **The CPU was designed specifically for this class ☺**

i281 CPU

- **The CPU was designed specifically for this class ☺**

- **It was designed by:**

  Kyung-Tae Kim and Alexander Stoytchev

i281 CPU Web Links

Download link for the software and hardware that comes with Bubble Sort preinstalled:

https://www.ece.iastate.edu/~alexs/classes/i281_CPU/

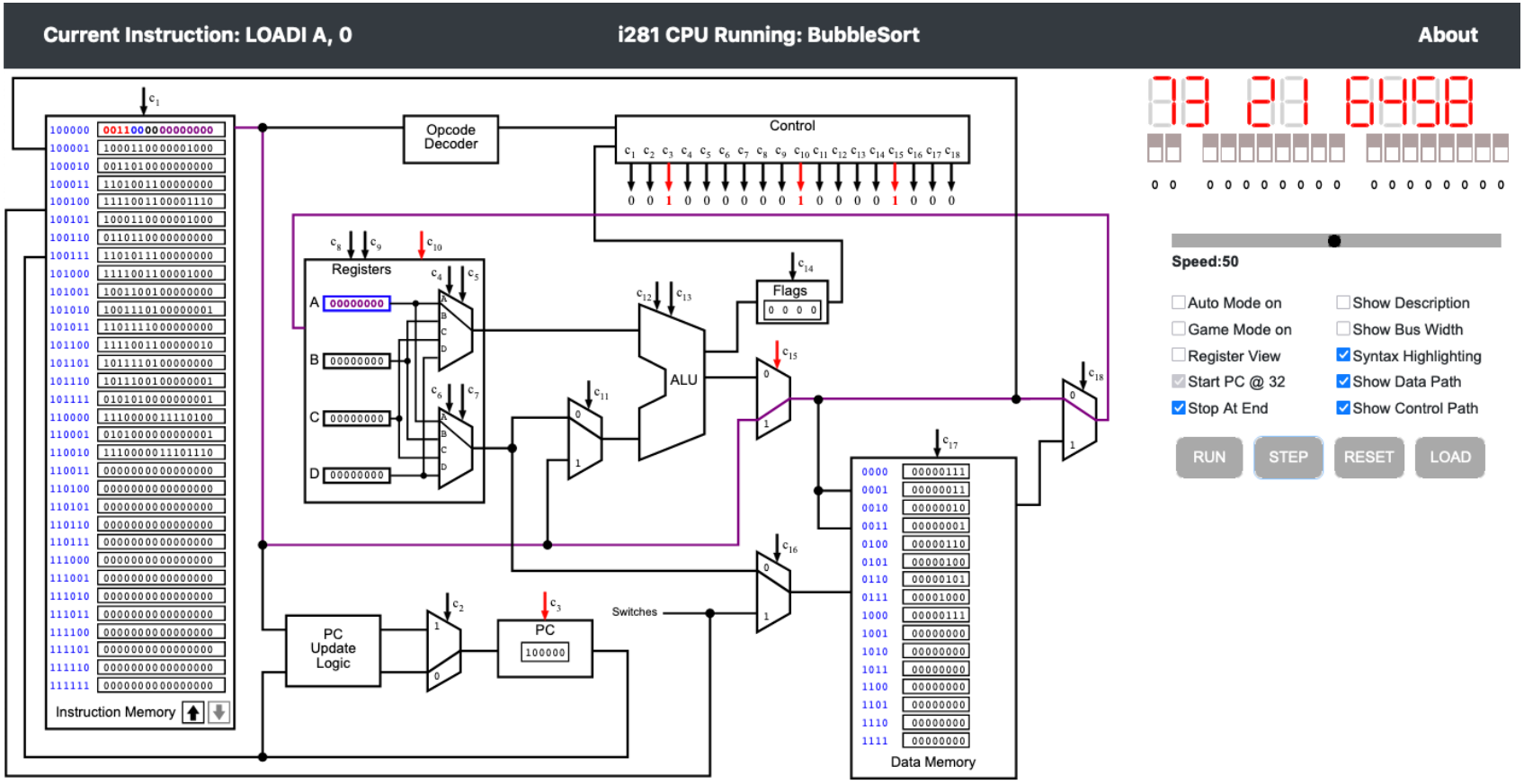
Download link for a ZIP file with the hardware version that comes with PONG preinstalled:

https://www.ece.iastate.edu/~alexs/classes/2025_Fall_2810/labs/Lab_13/

Web link for the i281 Simulator:

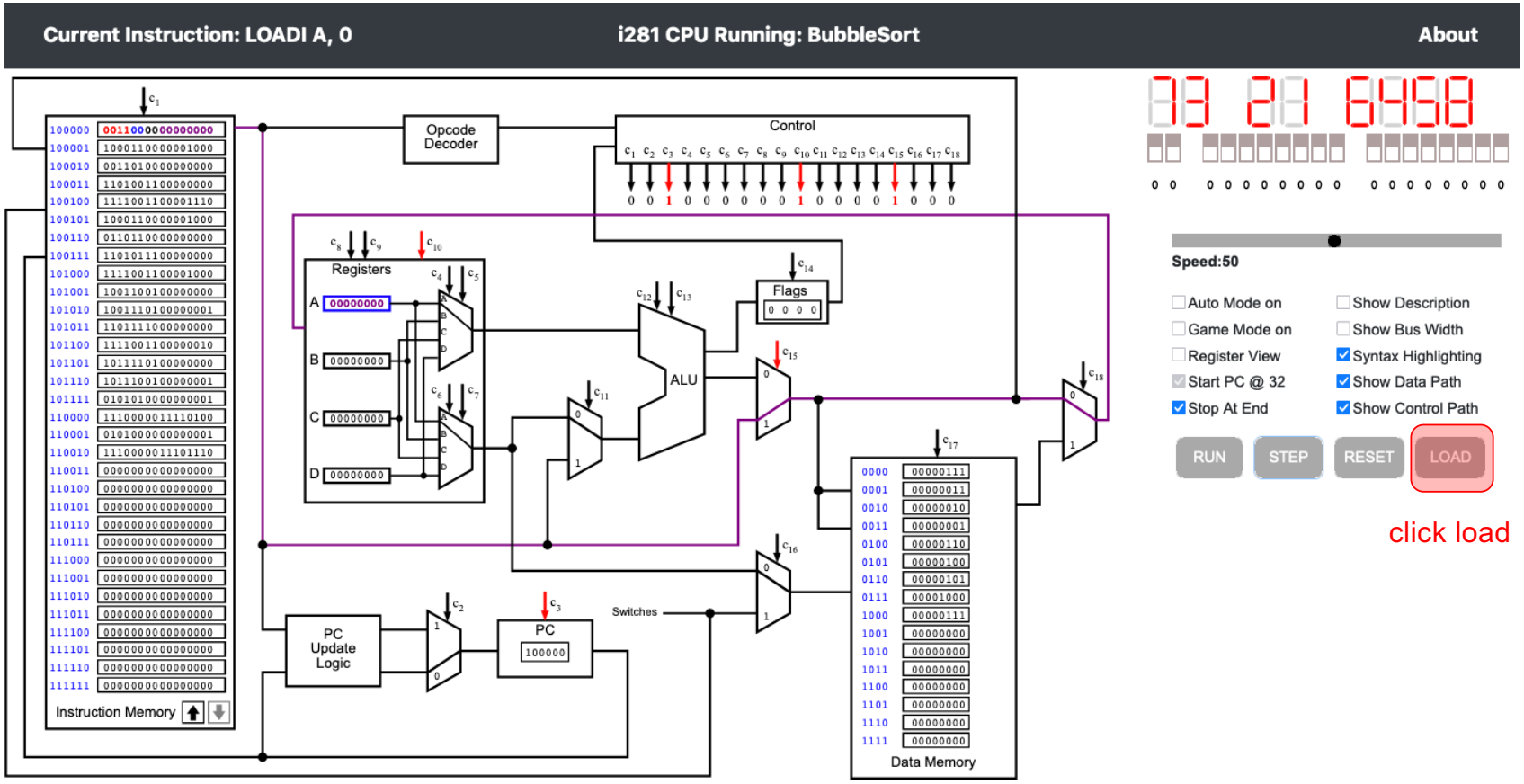
https://www.ece.iastate.edu/~alexs/classes/i281_simulator/index.html

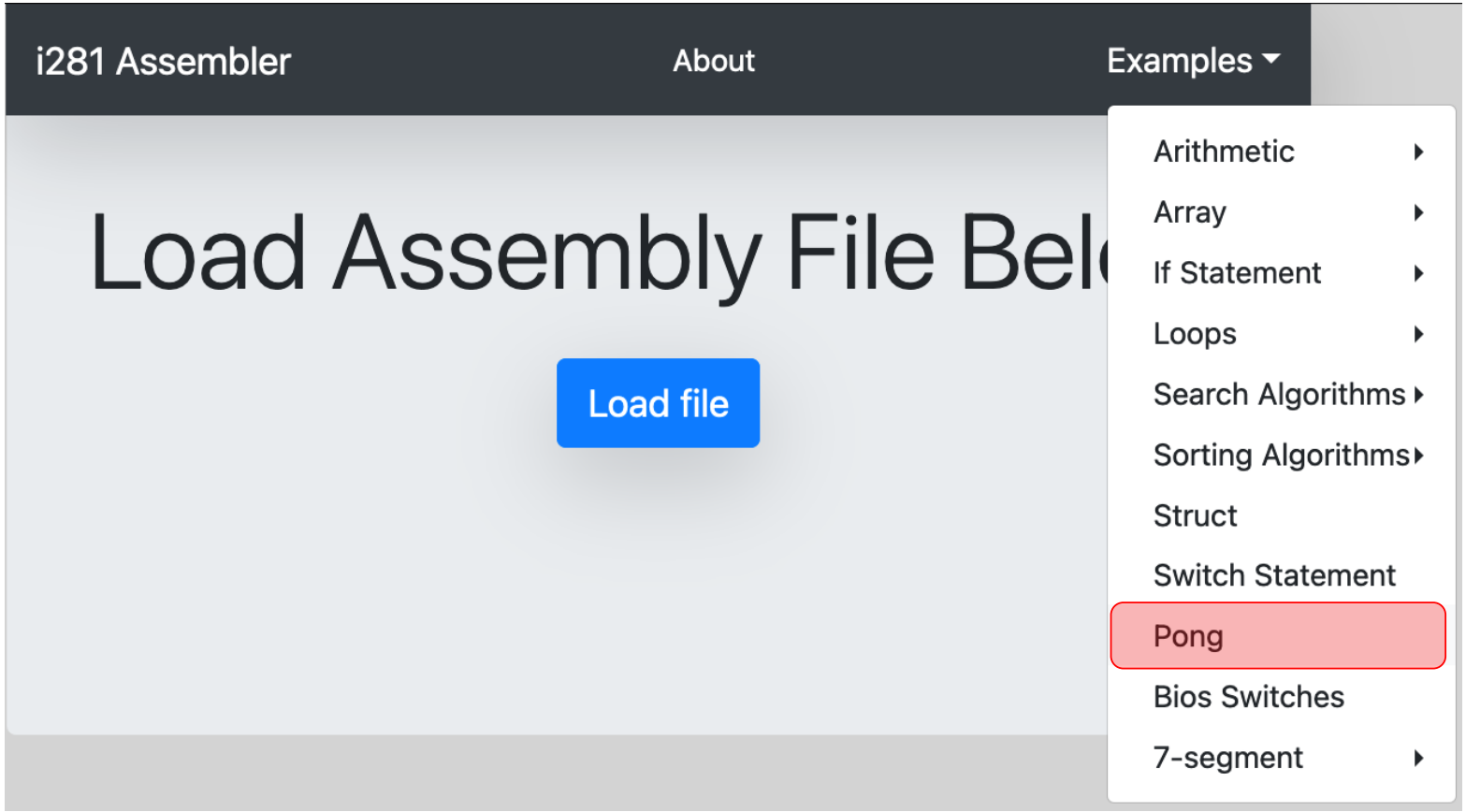
i281 Simulator



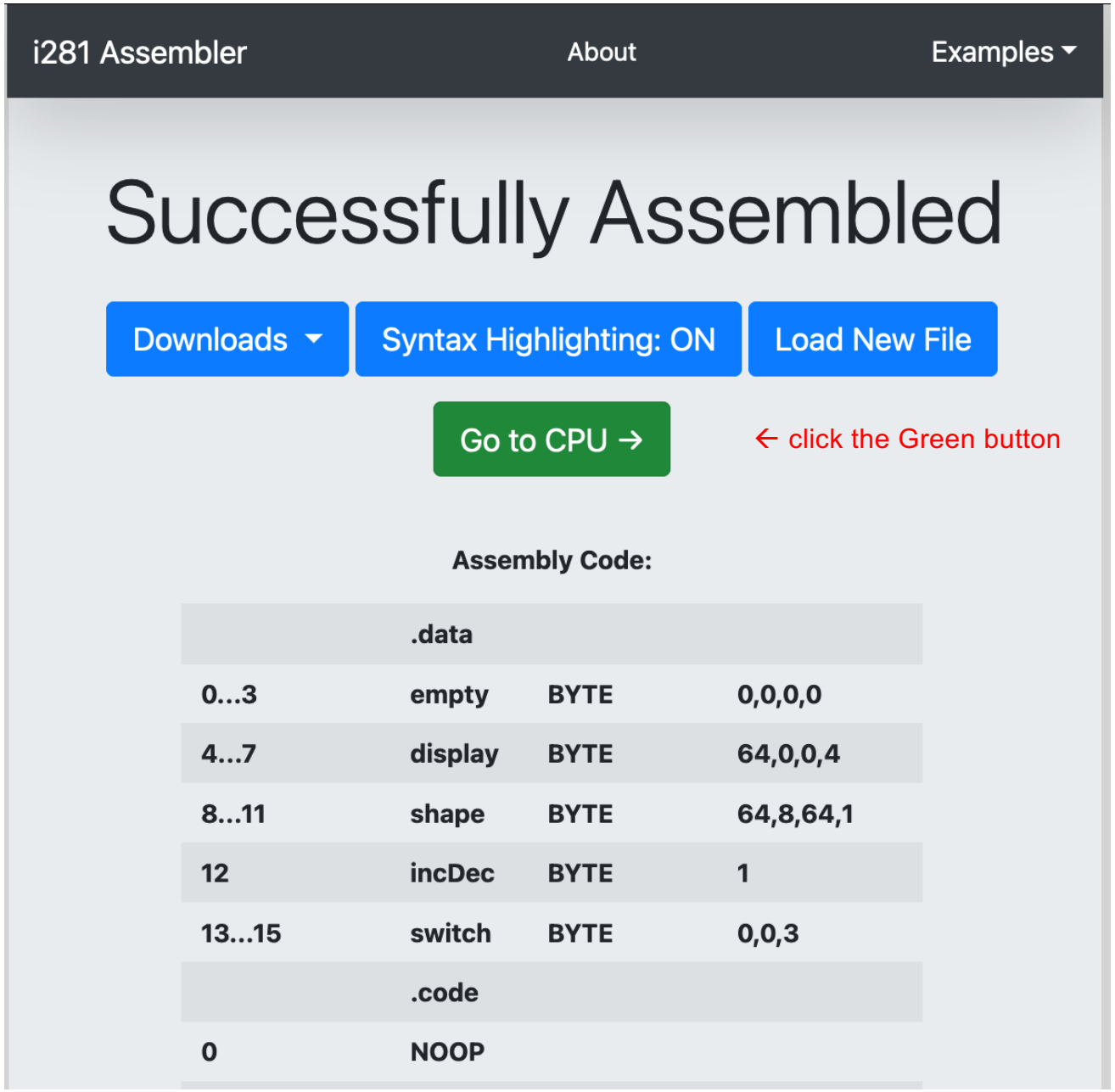
To try the simulator, go to the class web page and follow the link.

How to Play Pong in the Simulator

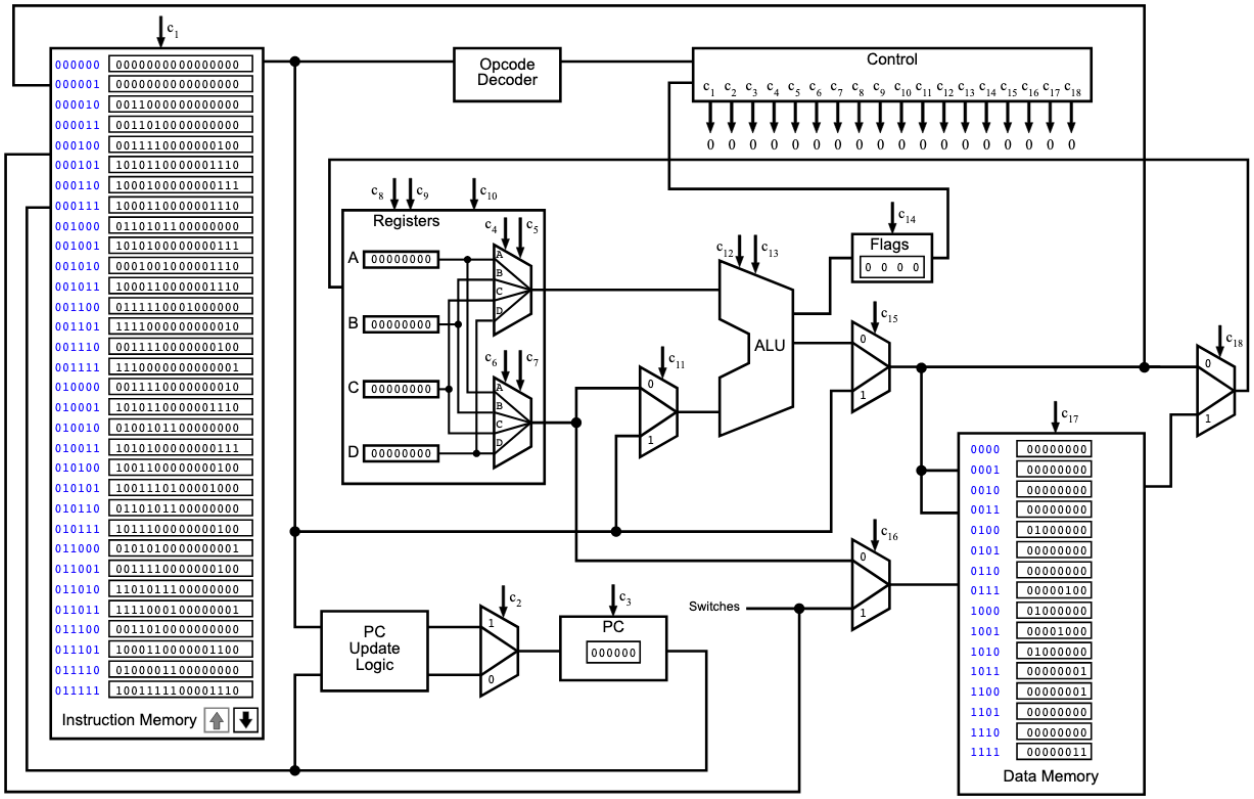


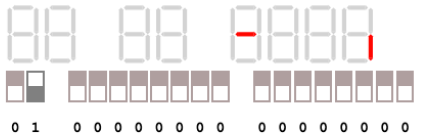


click on Examples and then select Pong


i281 Assembler
About
Examples
Successfully Assembled
Downloads
Syntax Highlighting: ON
Load New File
Go to CPU →
← click the Green button
Assembly Code:
.data
0...3 empty BYTE 0,0,0,0
4...7 display BYTE 64,0,0,4
8...11 shape BYTE 64,8,64,1
12 incDec BYTE 1
13...15 switch BYTE 0,0,3
.code
0 NOOP

Current Instruction: **i281 CPU Running: Pong** **About**



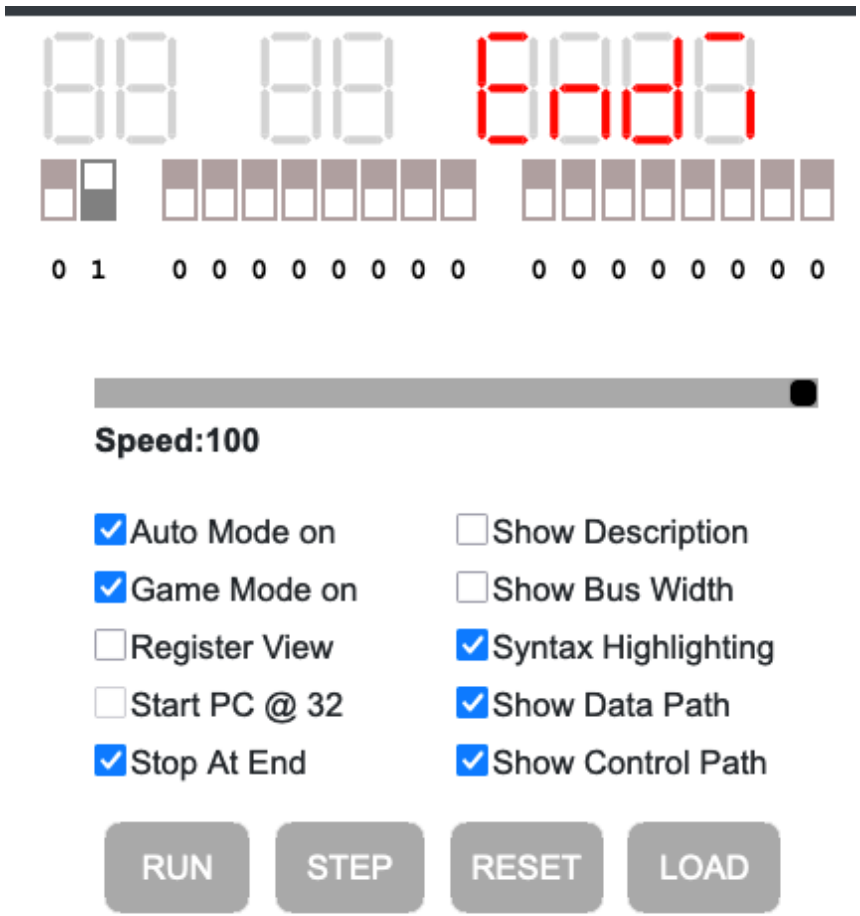


Speed:50

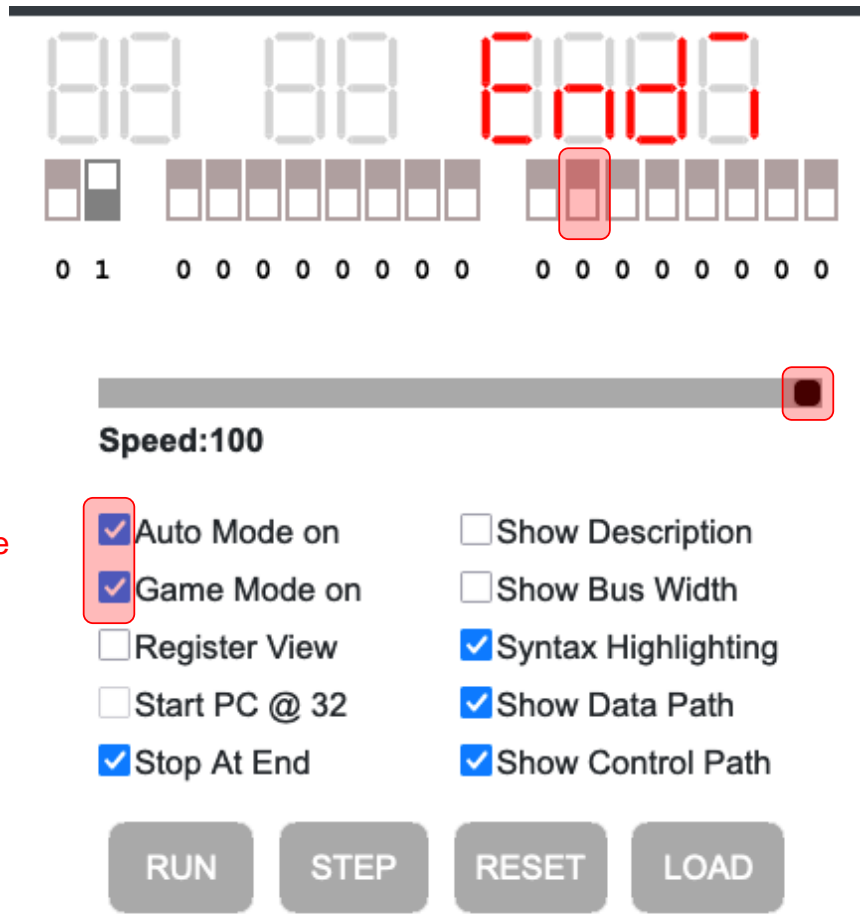
- [ ] Auto Mode on
- [x] Game Mode on
- [ ] Register View
- [ ] Start PC @ 32
- [x] Stop At End
- [ ] Show Description
- [ ] Show Bus Width
- [x] Syntax Highlighting
- [x] Show Data Path
- [x] Show Control Path

RUN STEP RESET LOAD

Game Mode in the Simulator



Game Mode in the Simulator

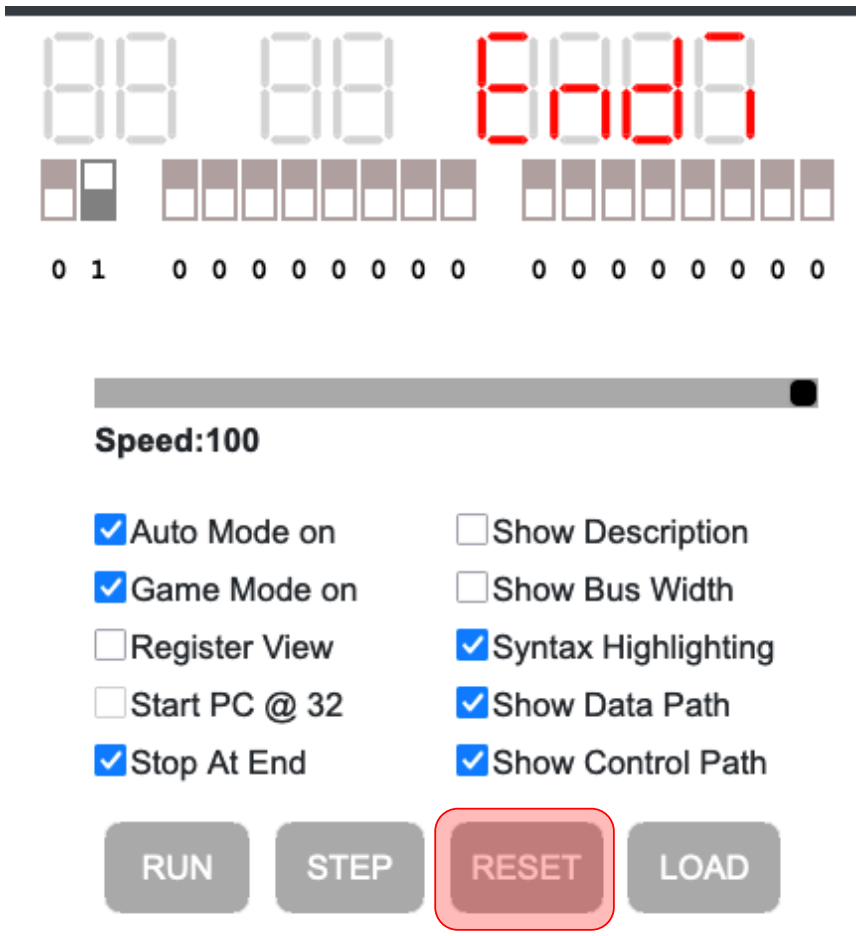


click on Switch 6 to move the paddle up/down

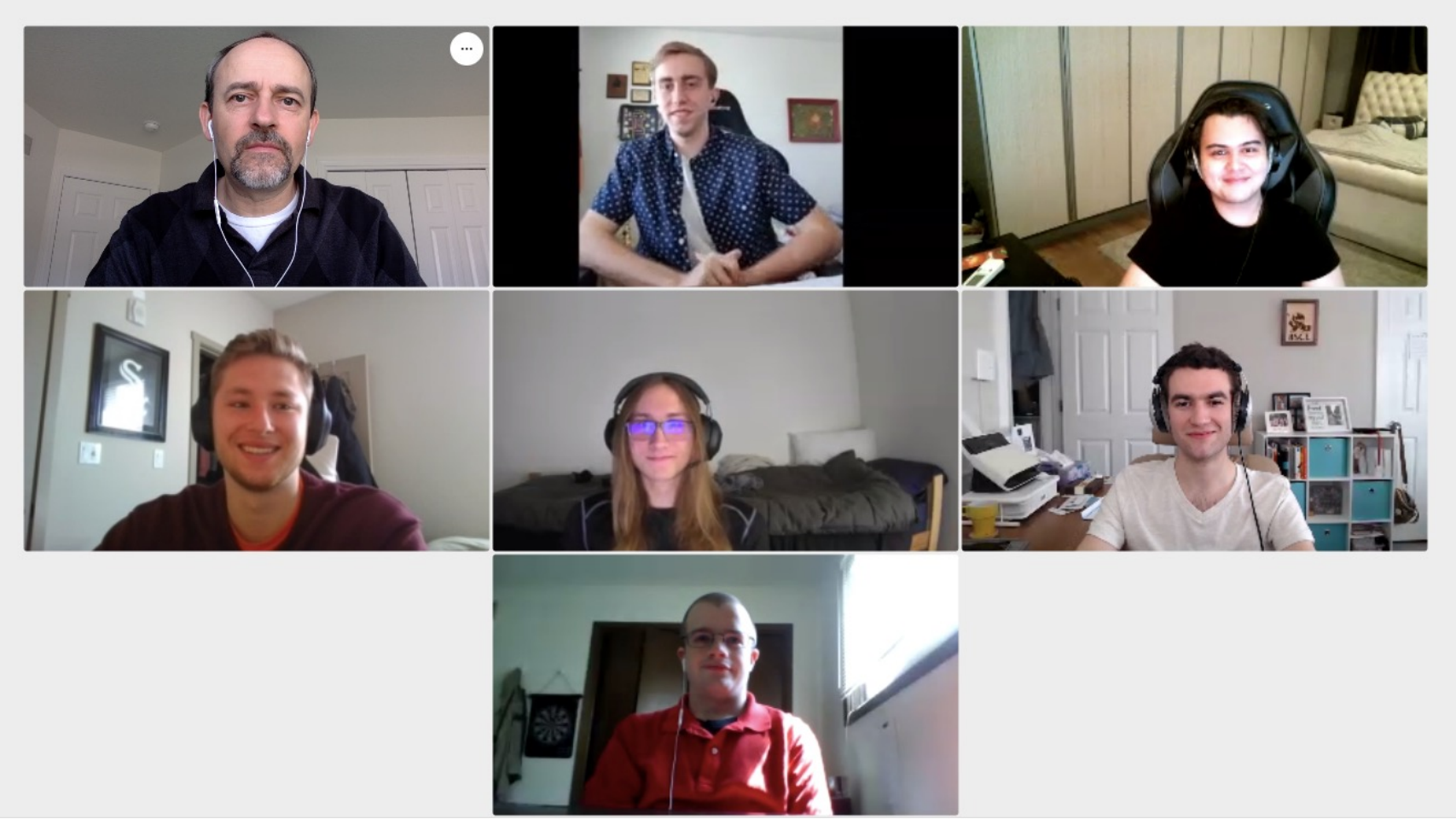
use the slider to increase the sped to 100

check these two boxes

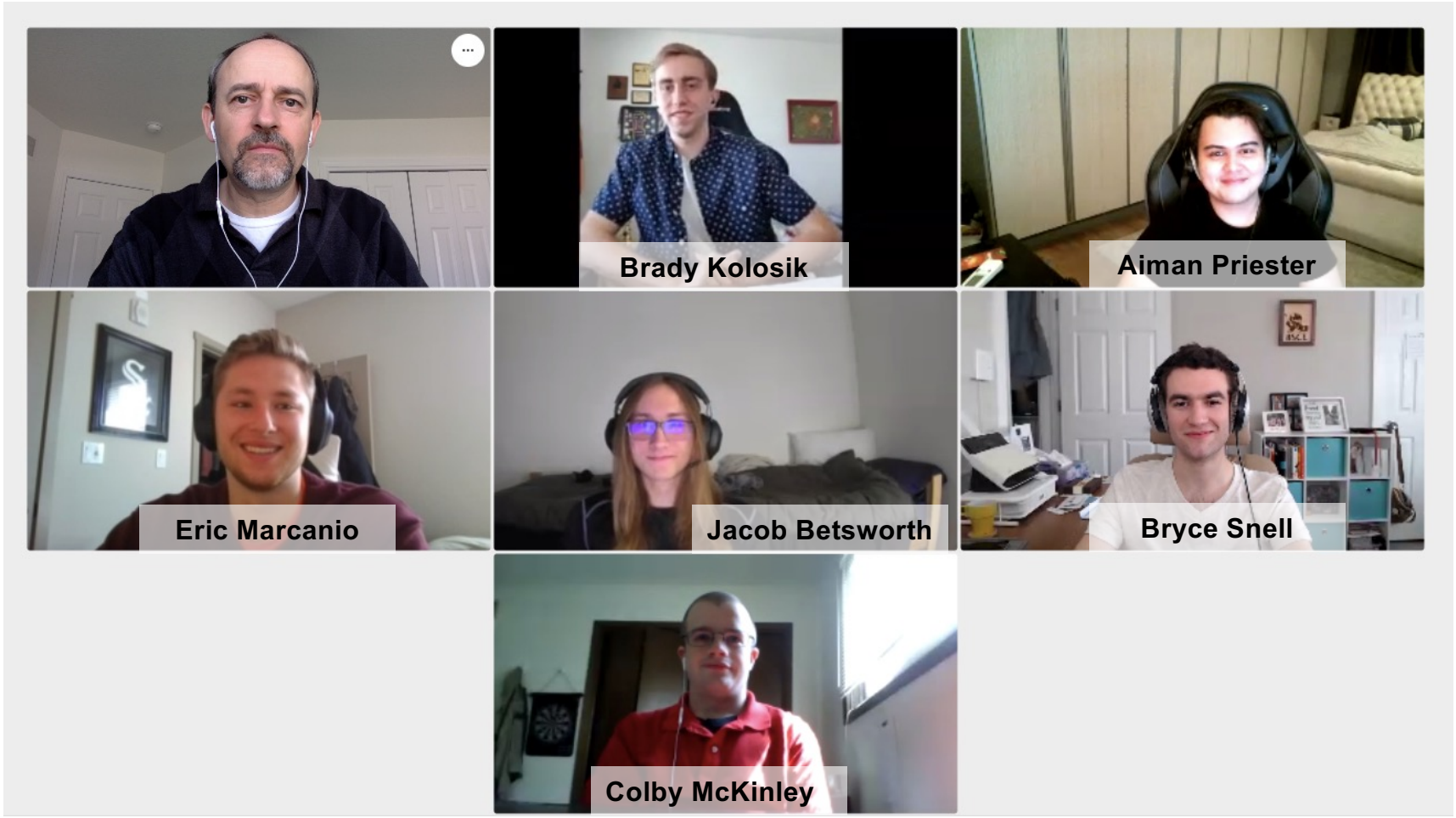
Game Mode in the Simulator



The i281 Simulator Was Implemented by a Senior Design Team (Fall 2020/Spring 2021)

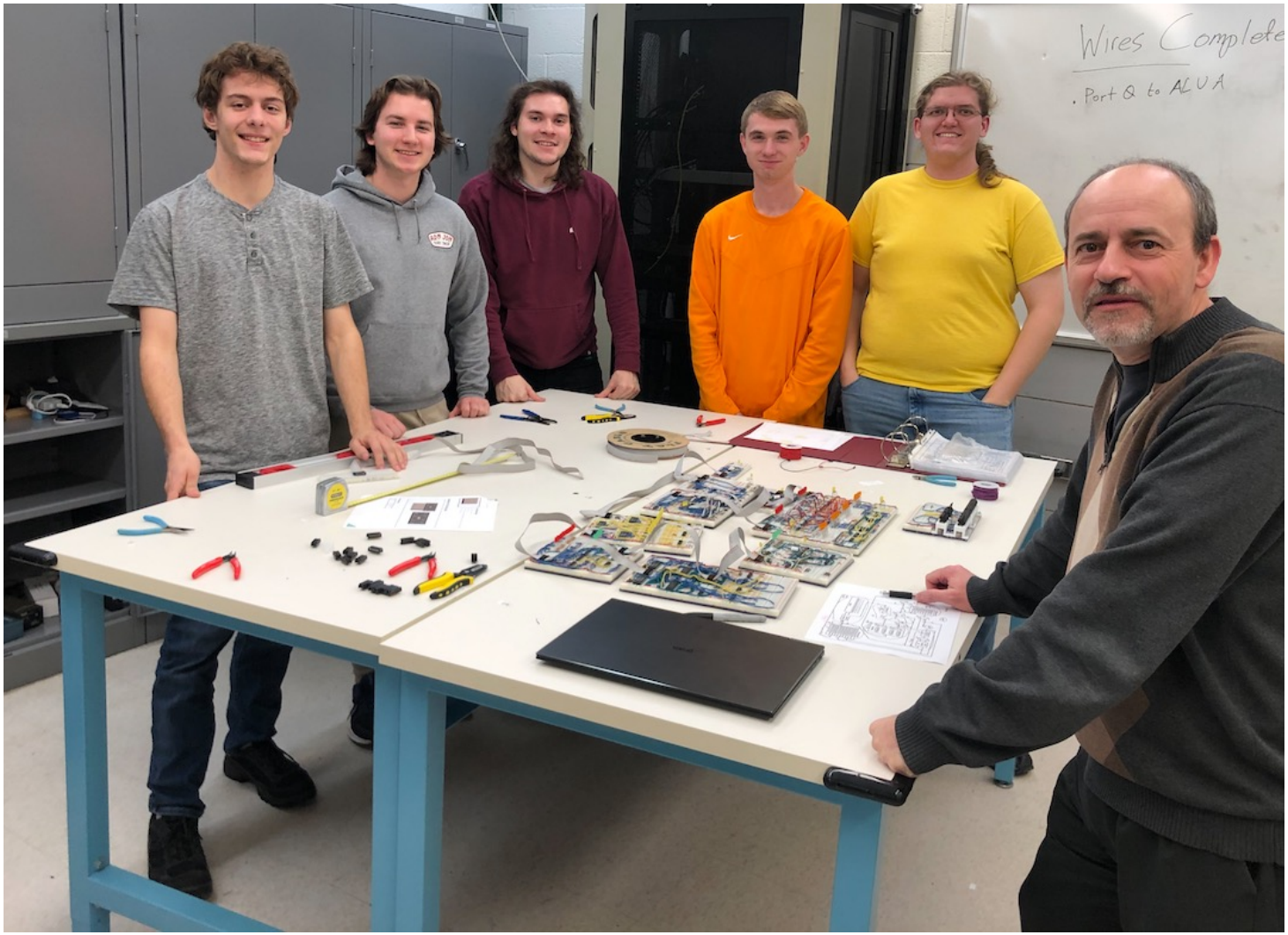


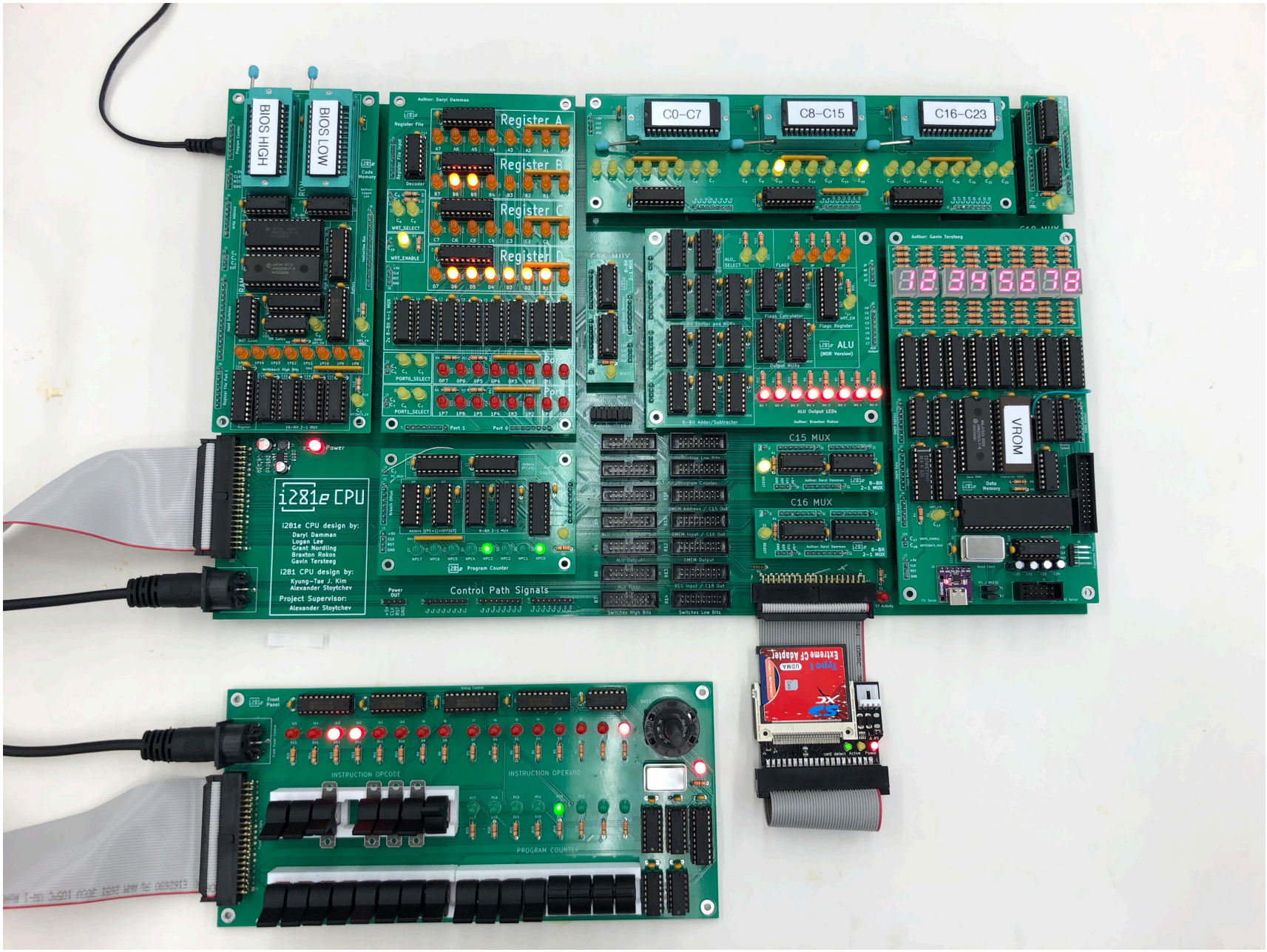
The i281 Simulator Was Implemented by a Senior Design Team (Fall 2020/Spring 2021)



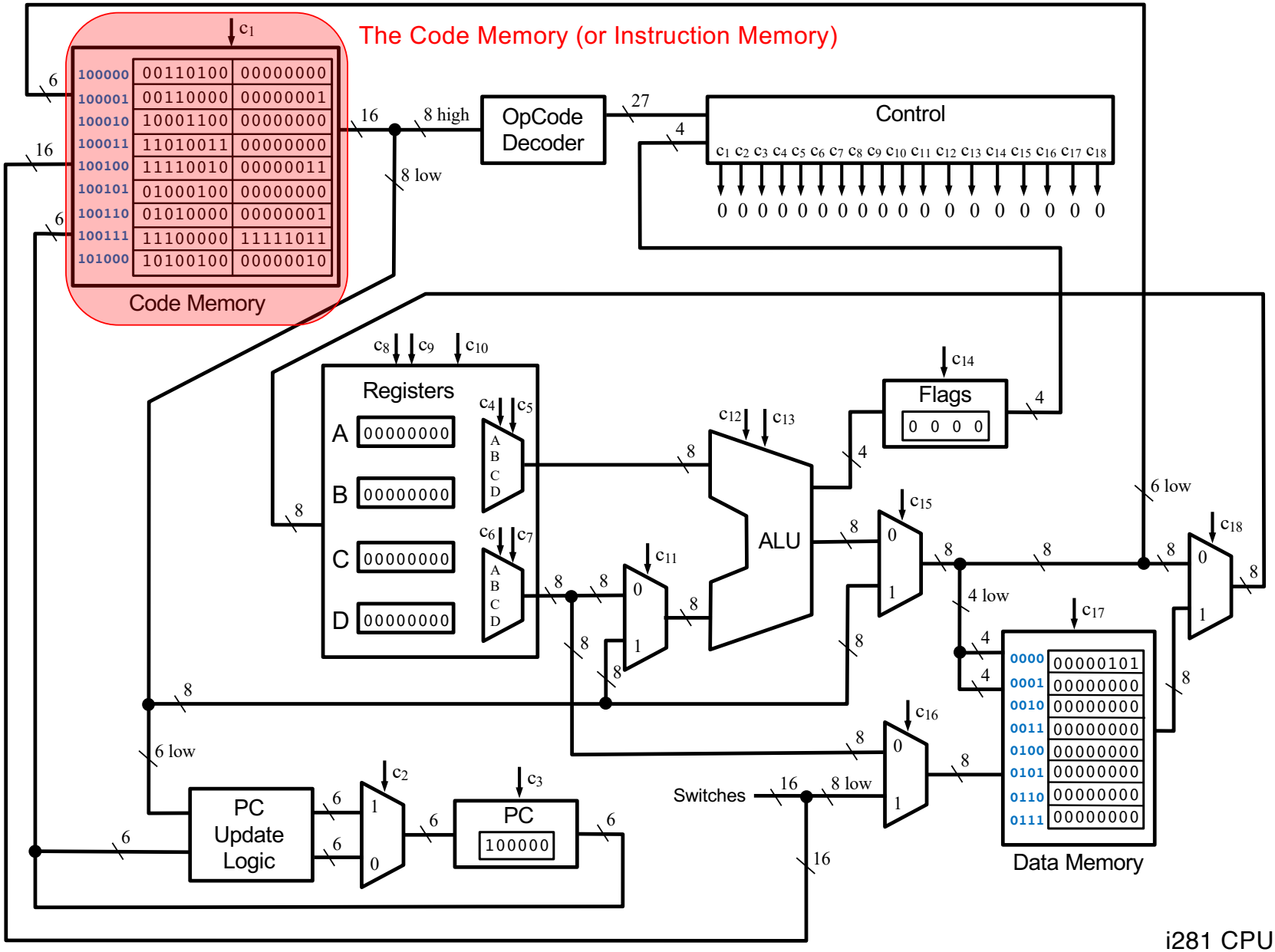
The i281e CPU

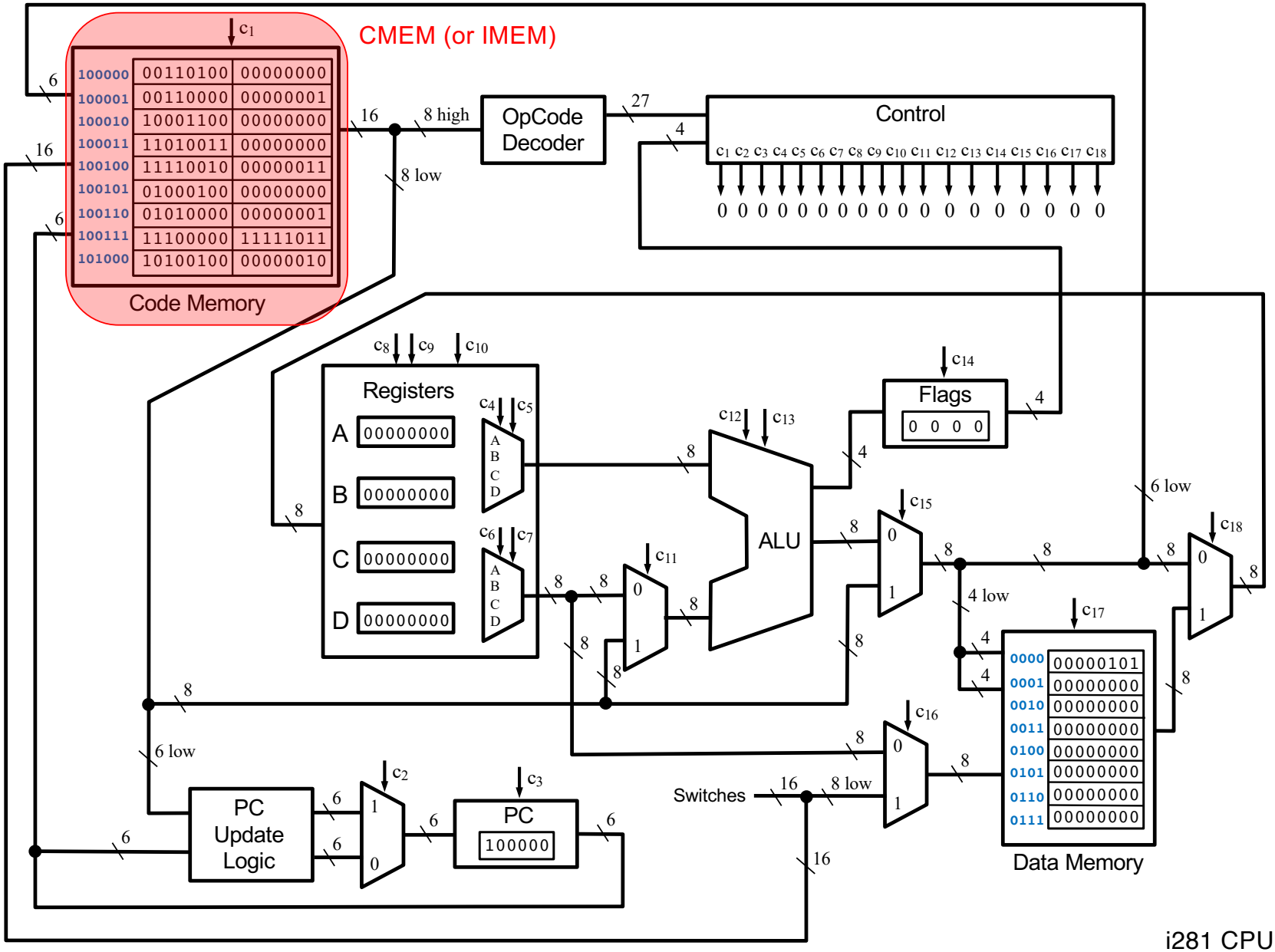
The hardware version of the i281 was implemented by another Senior Design Team (Fall 2023/Spring 2024)

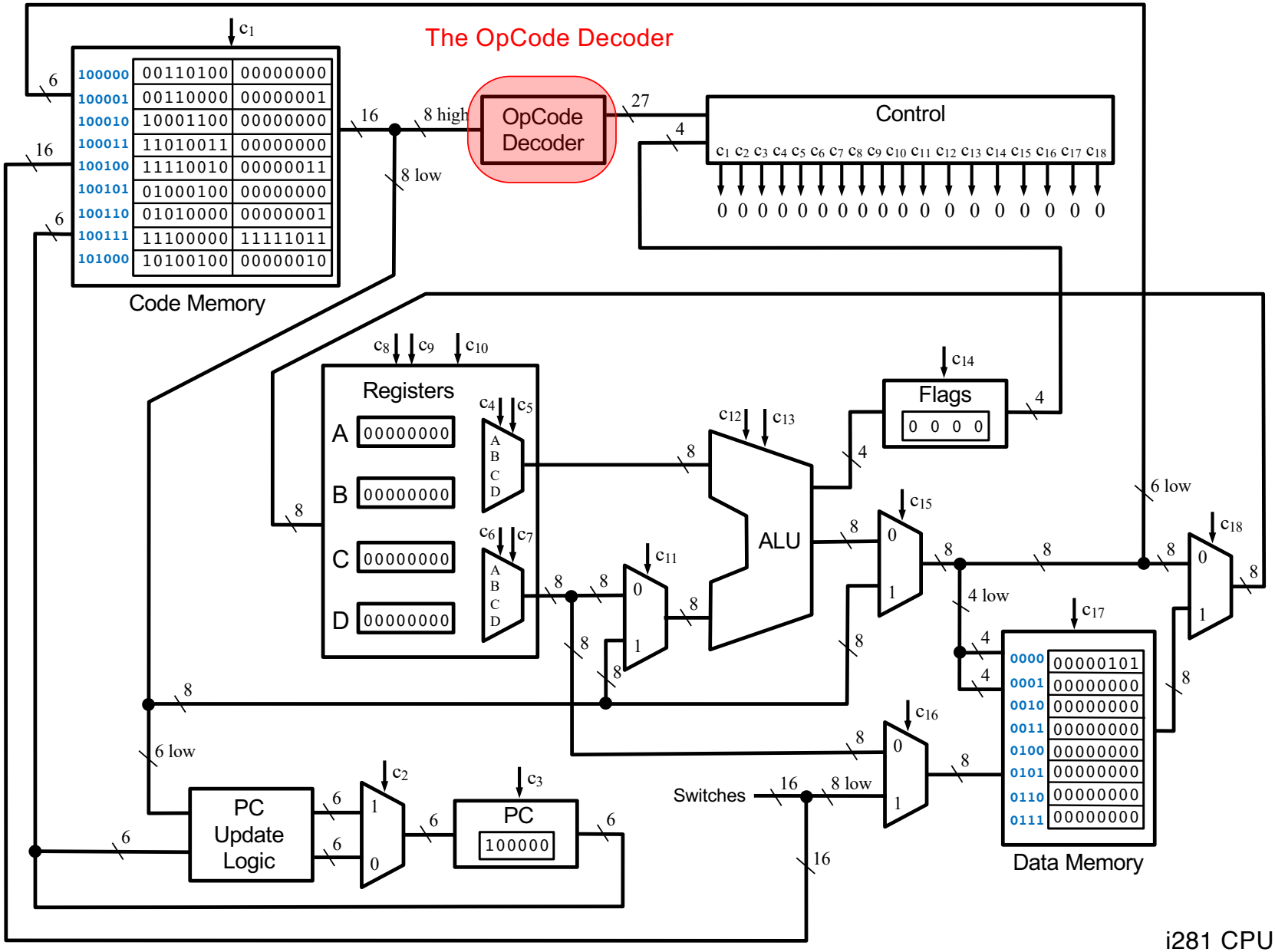


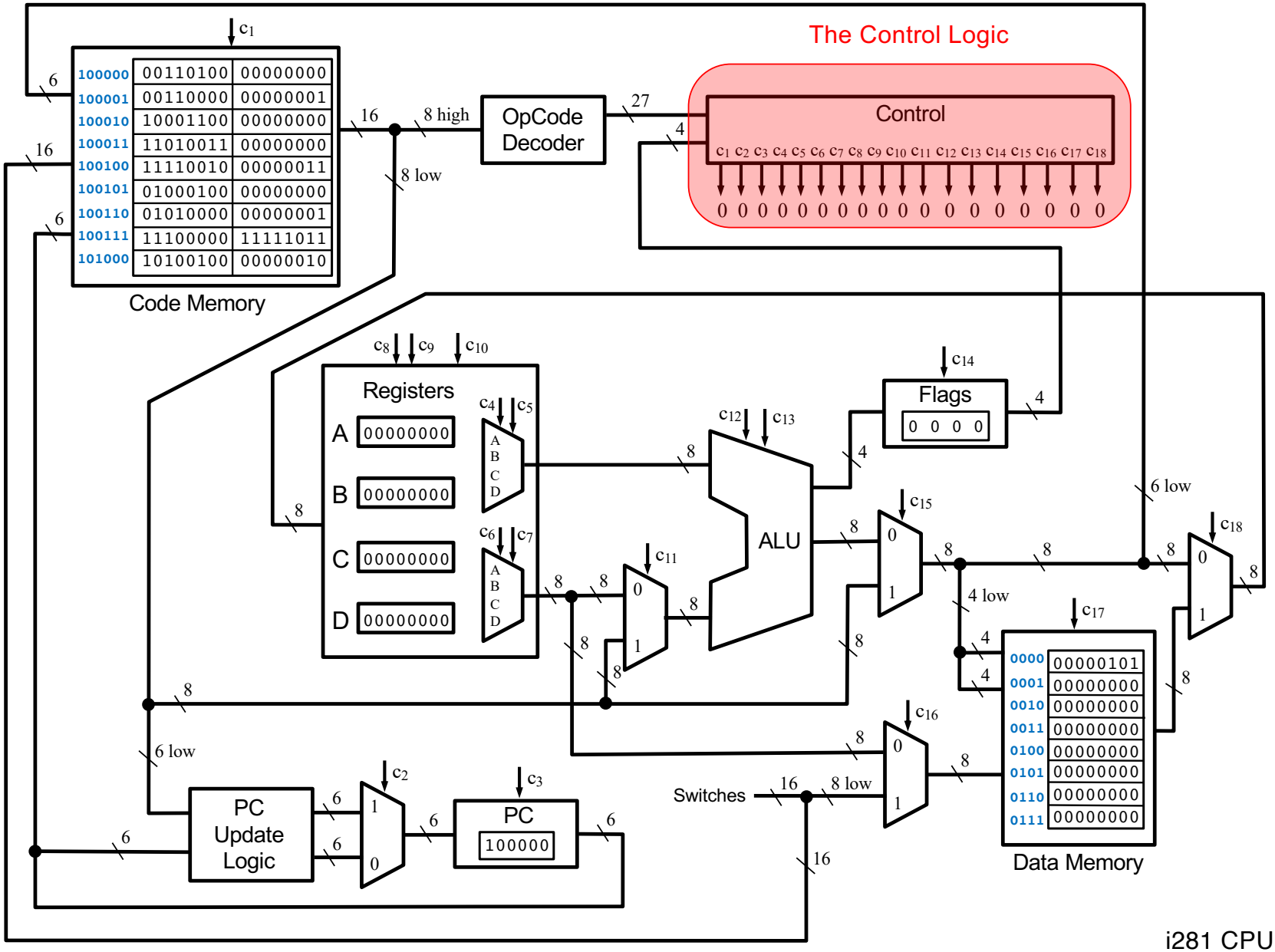

BIOS HIGH
BIOS LOW
Register A
Register B
Register C
Register D
C0–C7
C8–C15
C16–C23
C15 MUX
C16 MUX
VROM
ALU
Power
i281e CPU
i281e CPU design by:
Daryl Damman
Logan Lee
Grant Nordling
Braxton Rokos
Gavin Tersteeg
i281 CPU design by:
Kyung-Tae J. Kim
Alexander Stoytchev
Project Supervisor:
Alexander Stoytchev
Control Path Signals
Program Counter
INSTRUCTION OPCODE
INSTRUCTION OPERAND
PROGRAM COUNTER
Extreme CF Adapter

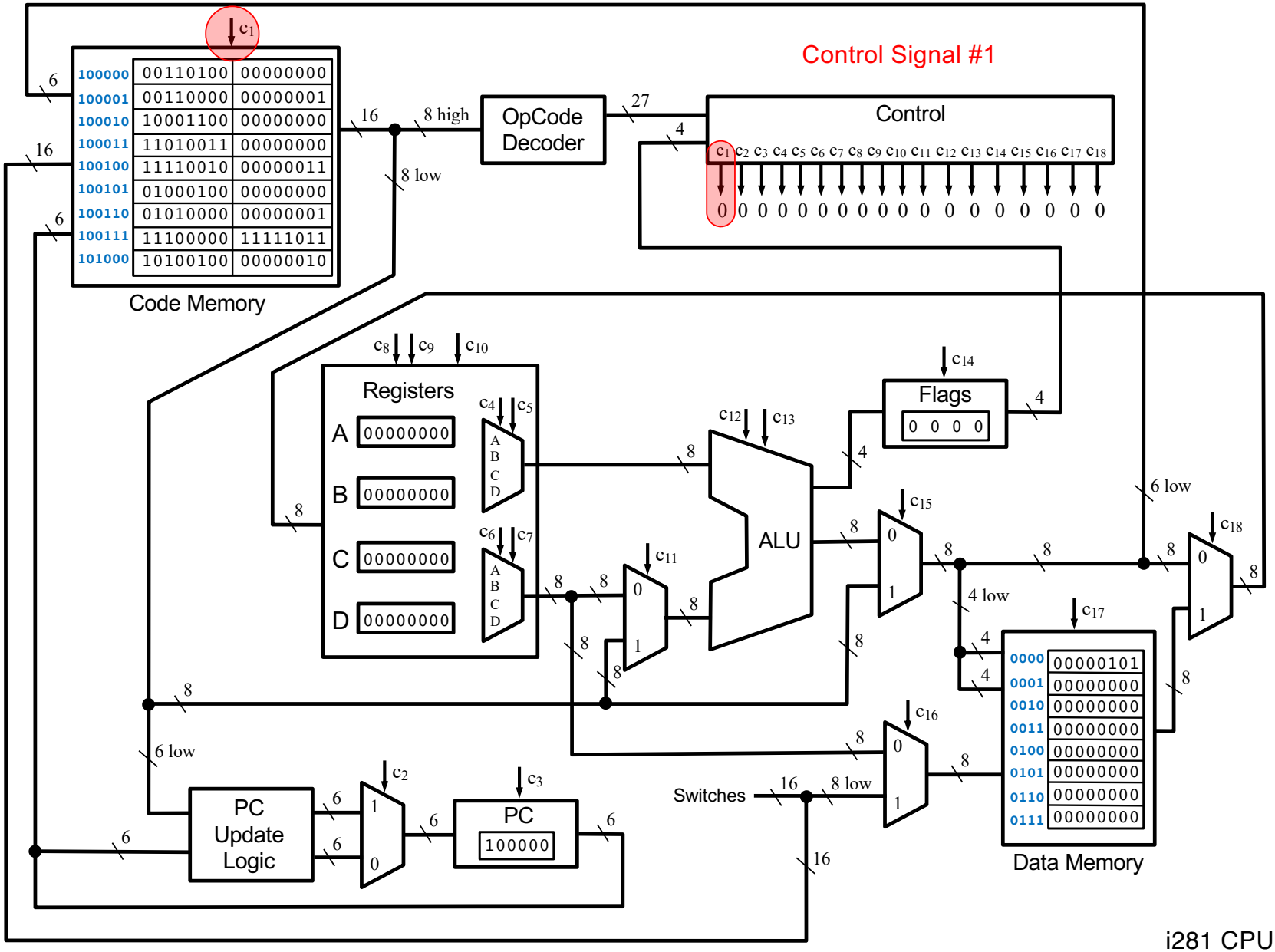
The CPU Components

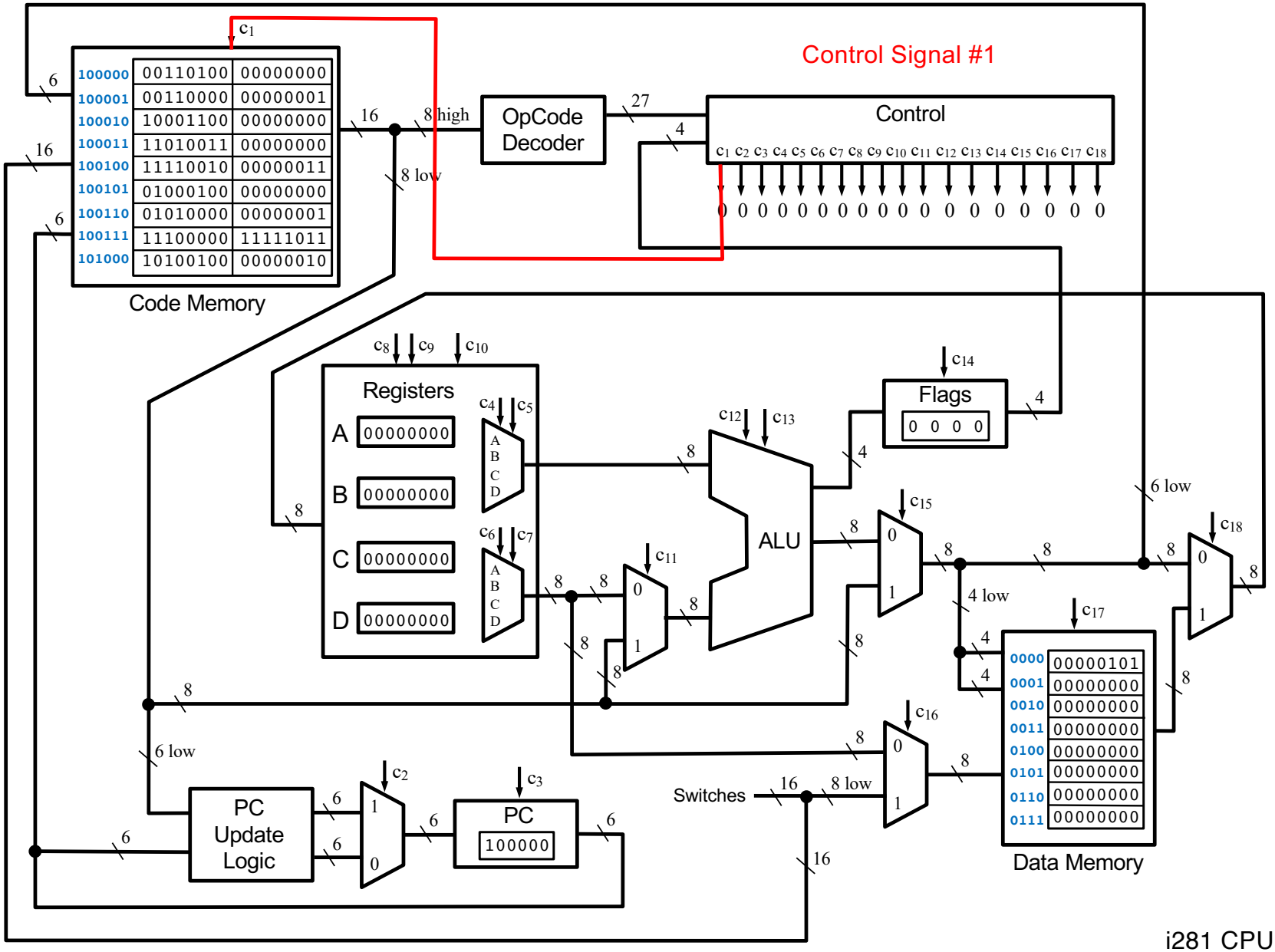

The Code Memory (or Instruction Memory)
c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
c14
Flags
0 0 0 0
4
c12 c13
ALU
c15
c11
c16
c17
c18
6 low
4 low
8 low
8 high
Switches
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
PC Update Logic
c2
c3
PC
100000
i281 CPU

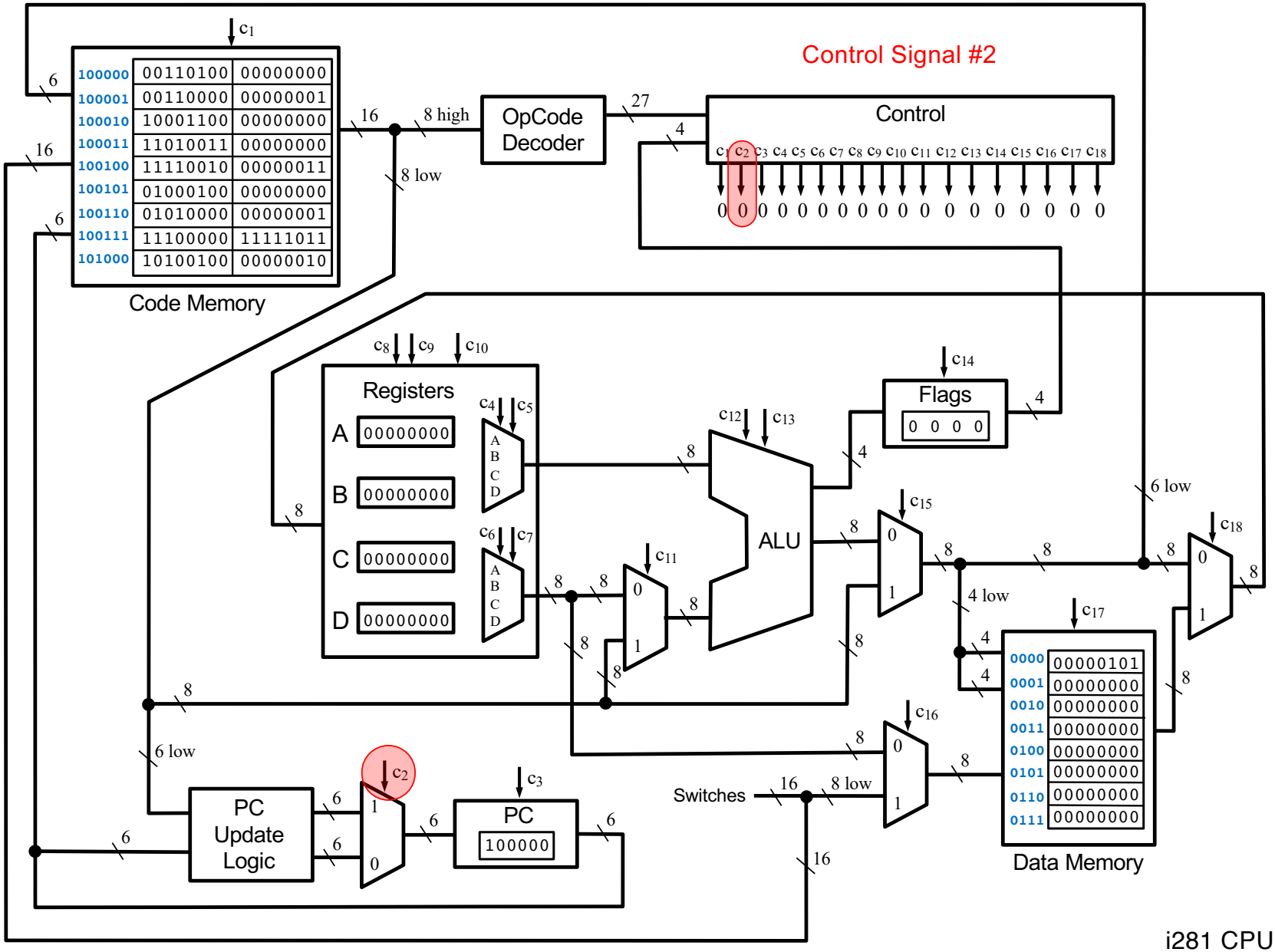

CMEM (or IMEM)
c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
c14
Flags
0 0 0 0
4
c12 c13
ALU
c15
c11
c16
c17
c18
6 low
4 low
8 low
Switches
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
PC Update Logic
c2
c3
PC
100000
i281 CPU

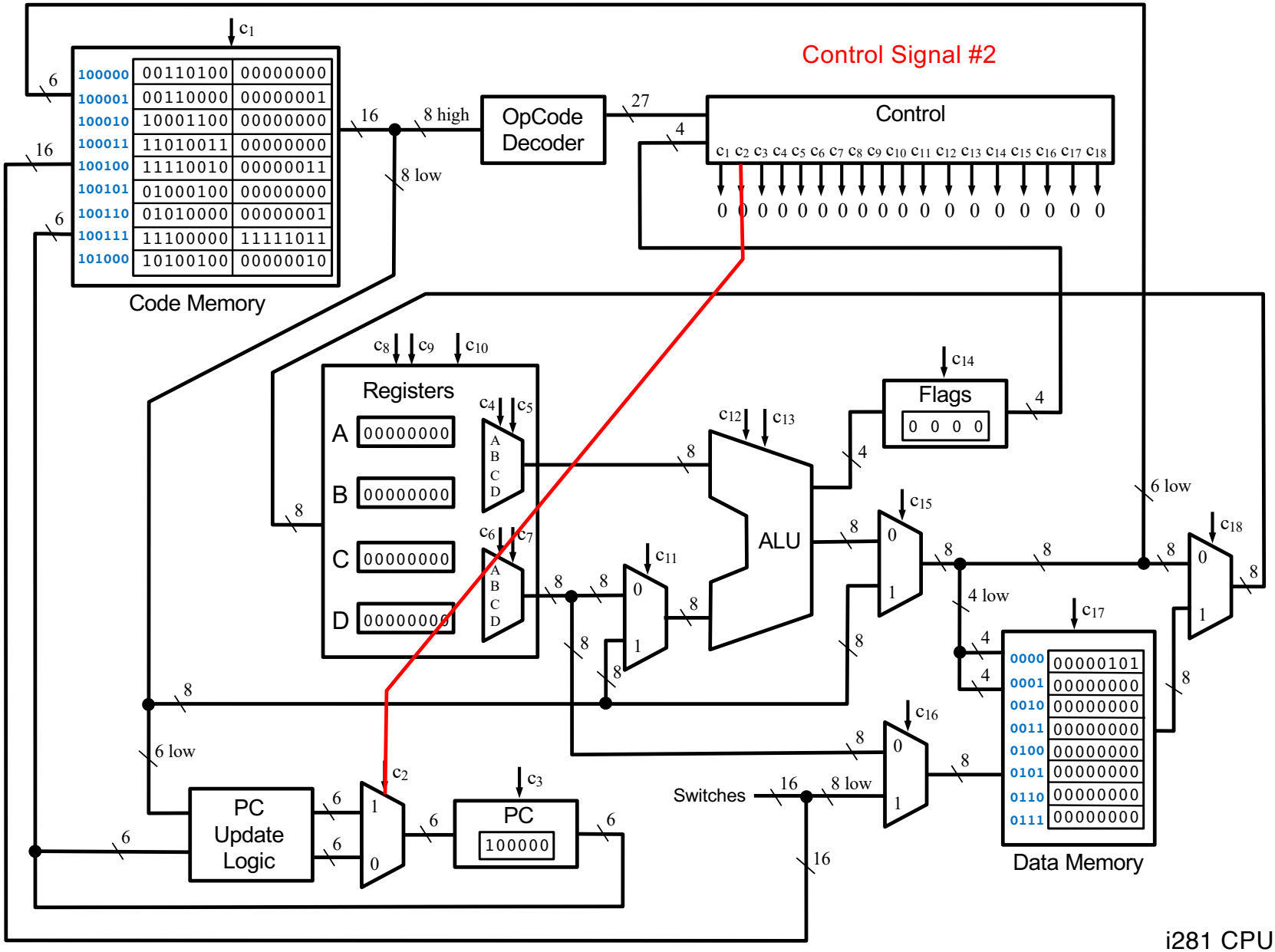

The OpCode Decoder
c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c12 c13
ALU
c14
Flags
0 0 0 0
4
c15
c11
c16
c17
c18
6 low
4 low
8 low
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
Switches
16
PC Update Logic
c2
c3
PC
100000
i281 CPU

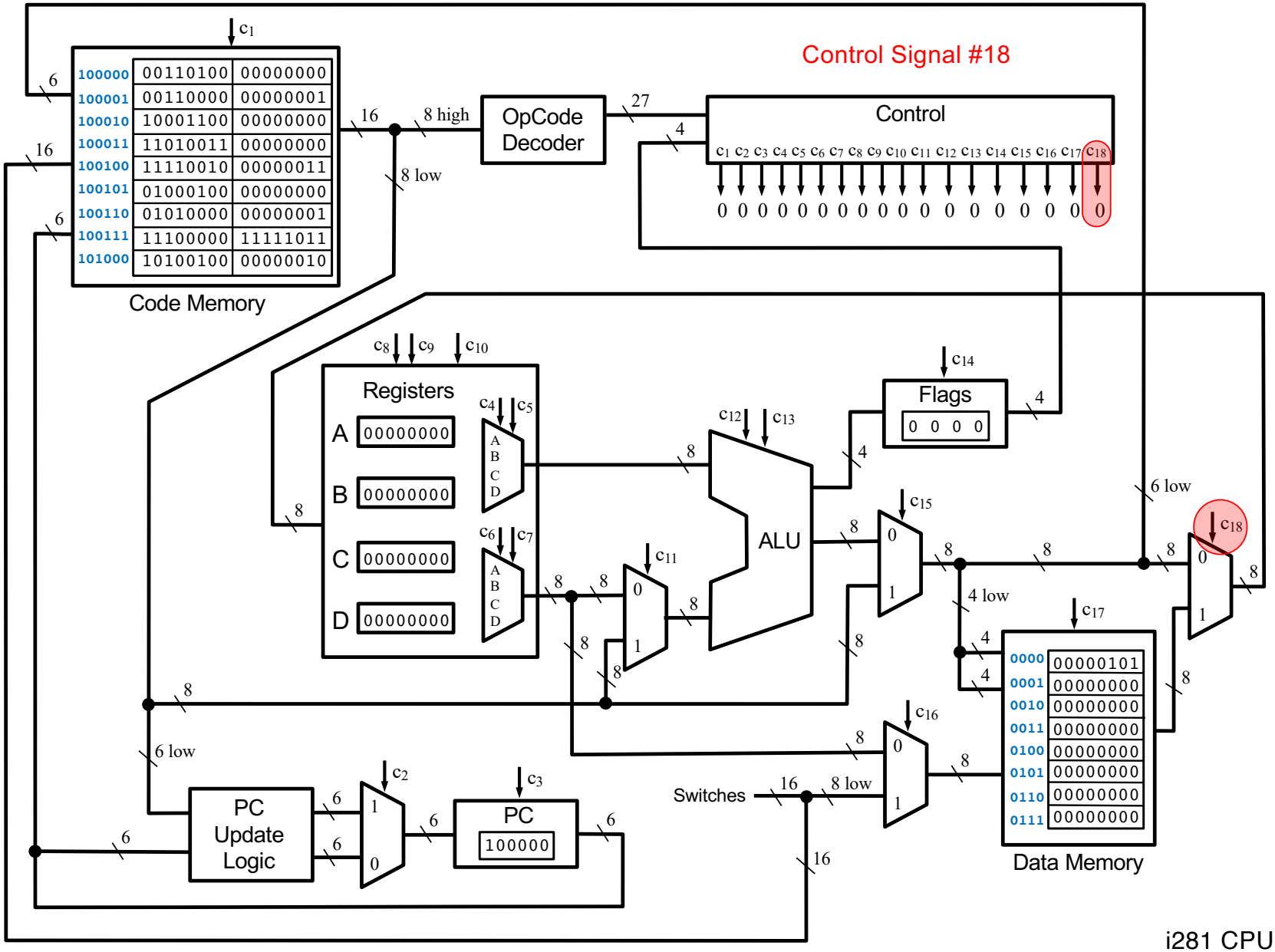

The Control Logic
c1
Code Memory
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
c14
Flags
0 0 0 0
c12 c13
ALU
c15
c11
c16
c17
c18
6 low
4 low
8 low
Switches
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
PC Update Logic
c2
c3
PC
100000
i281 CPU

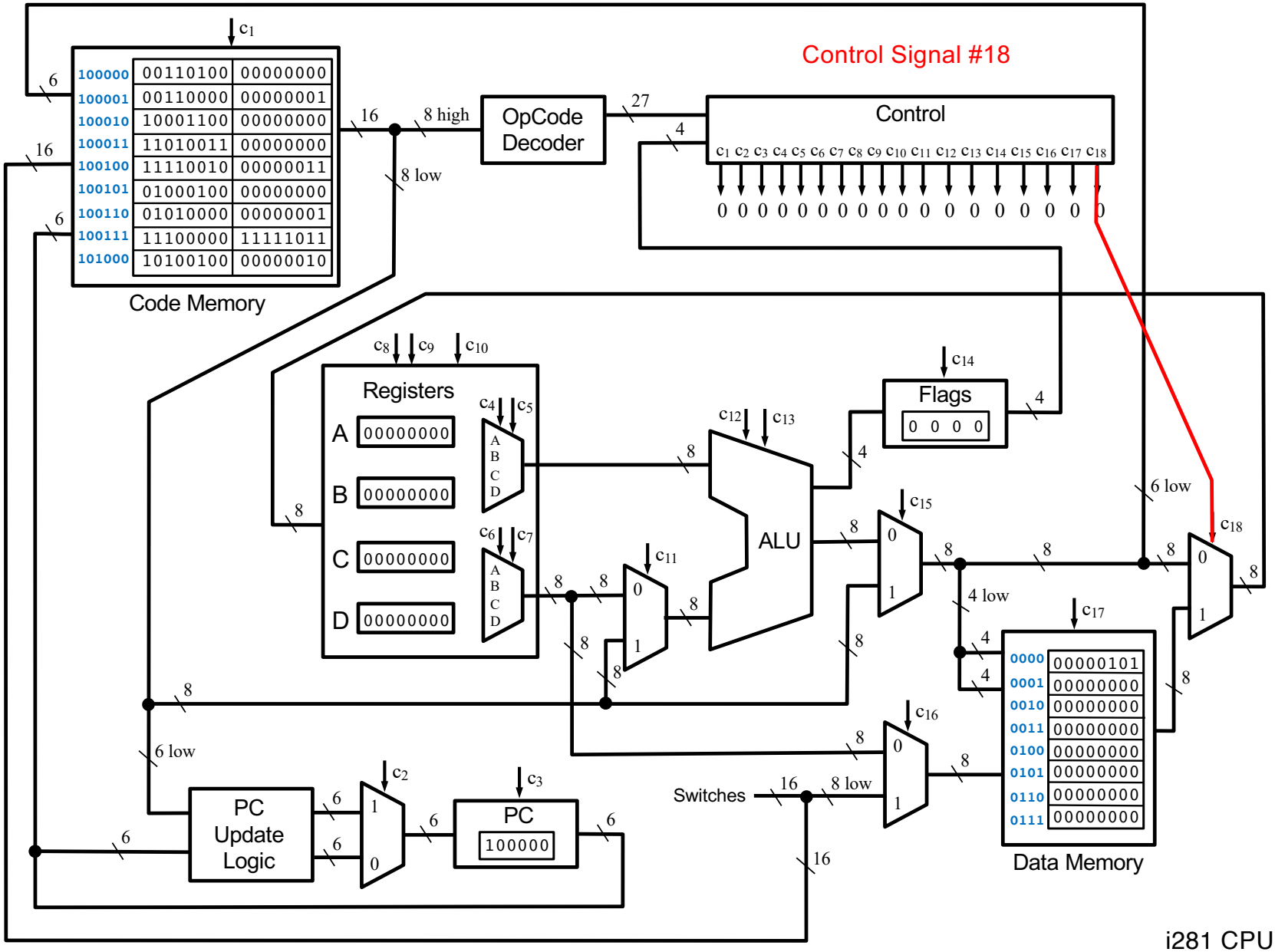

Control Signal #1
c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Registers
c8 c9 c10
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c11
c12 c13
ALU
c14
Flags
0 0 0 0
4
c15
c16
c17
c18
6 low
4 low
8 low
Switches
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
PC Update Logic
c2
c3
PC
100000
i281 CPU

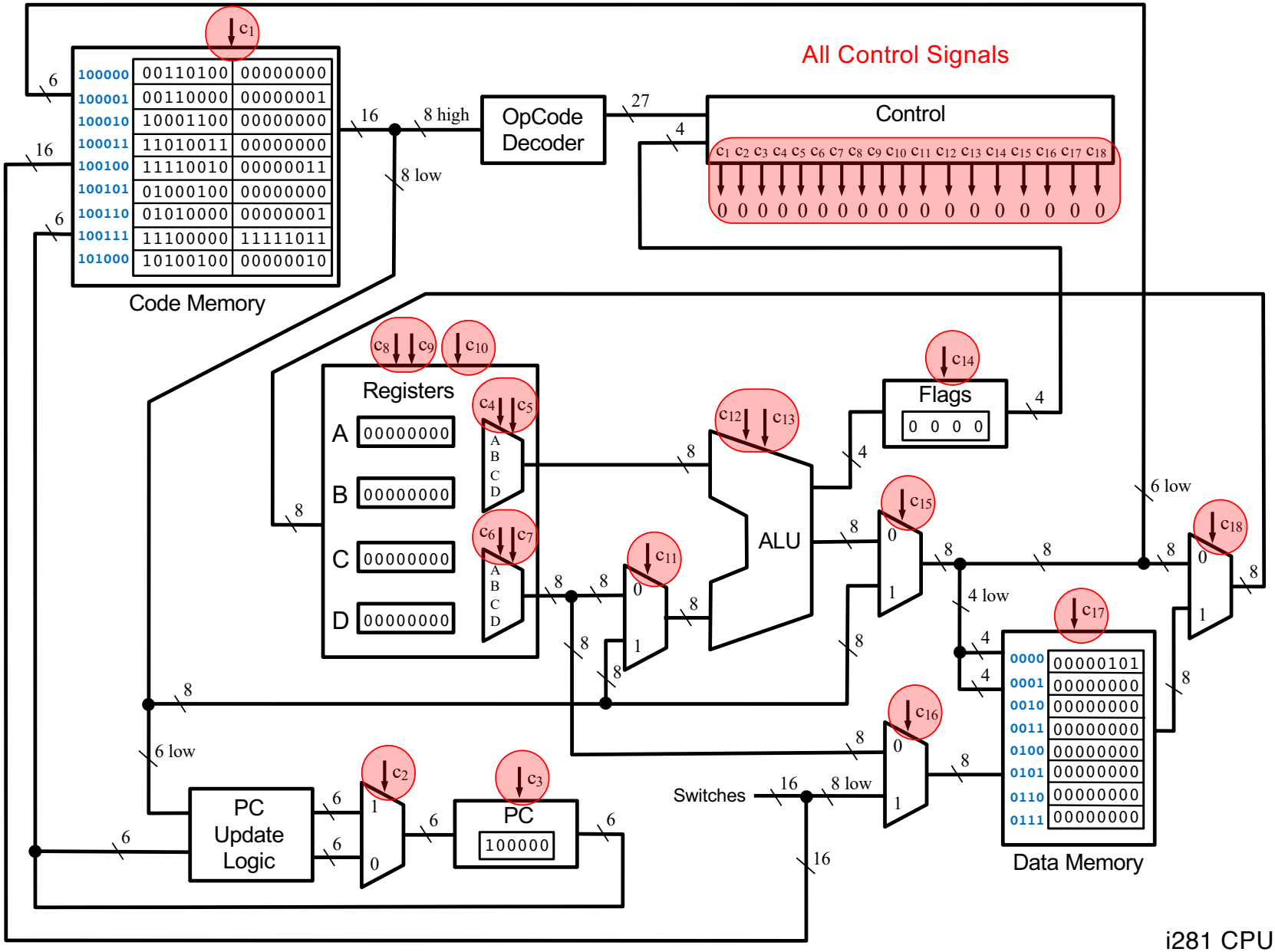

Control Signal #1
c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c8 c9 c10
c4 c5
c6 c7
A B C D
c11
c12 c13
ALU
c14
Flags
0 0 0 0
c15
c16
c17
c18
6 low
4 low
Switches
8 low
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
PC Update Logic
c2
c3
PC
100000
i281 CPU

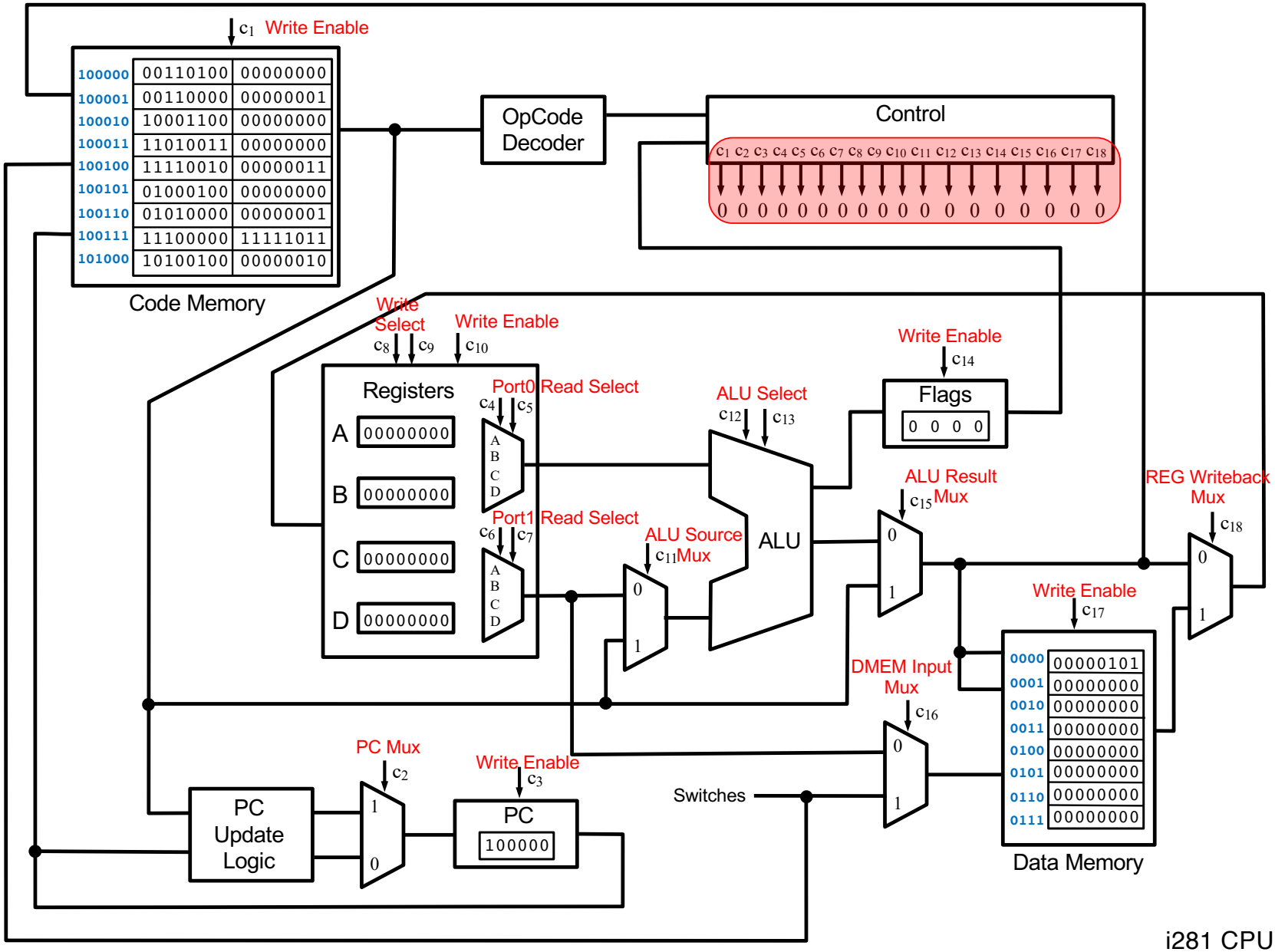

Control Signal #2
c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Registers
c8 c9 c10
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5 c6 c7
A B C D
c14
Flags
0 0 0 0
c12 c13
ALU
c15
c11
c16
c17
c18
6 low
4 low
Switches
8 low
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
c2
c3
PC Update Logic
PC
100000
i281 CPU

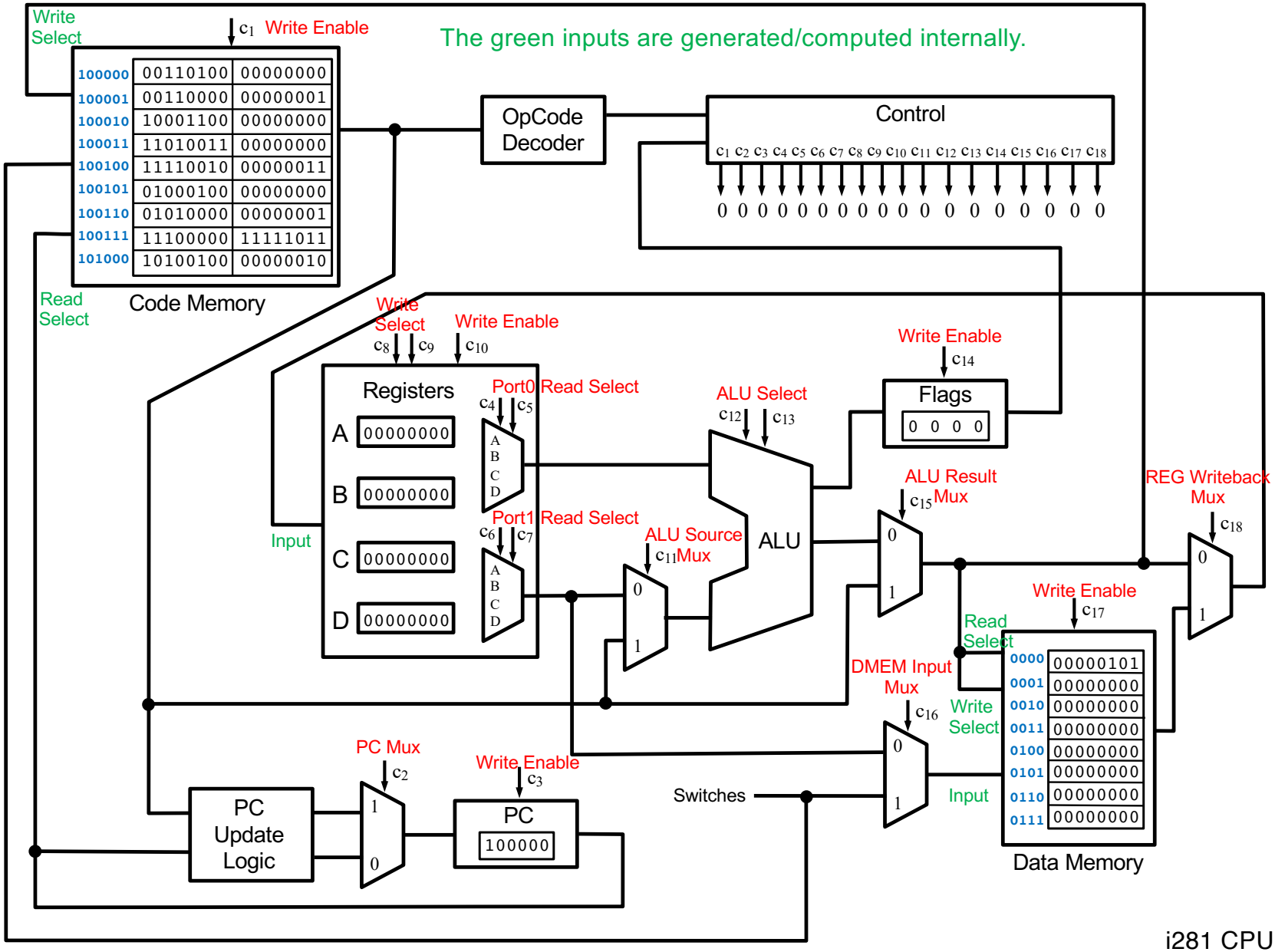

Control Signal #2
c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c8 c9 c10
c4 c5 c6 c7
A B C D
ALU
c12 c13
Flags
0 0 0 0
c14
c15
c11
c16
c17
c18
6 low
4 low
8 low
Switches
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
PC Update Logic
PC
100000
c2
c3
i281 CPU

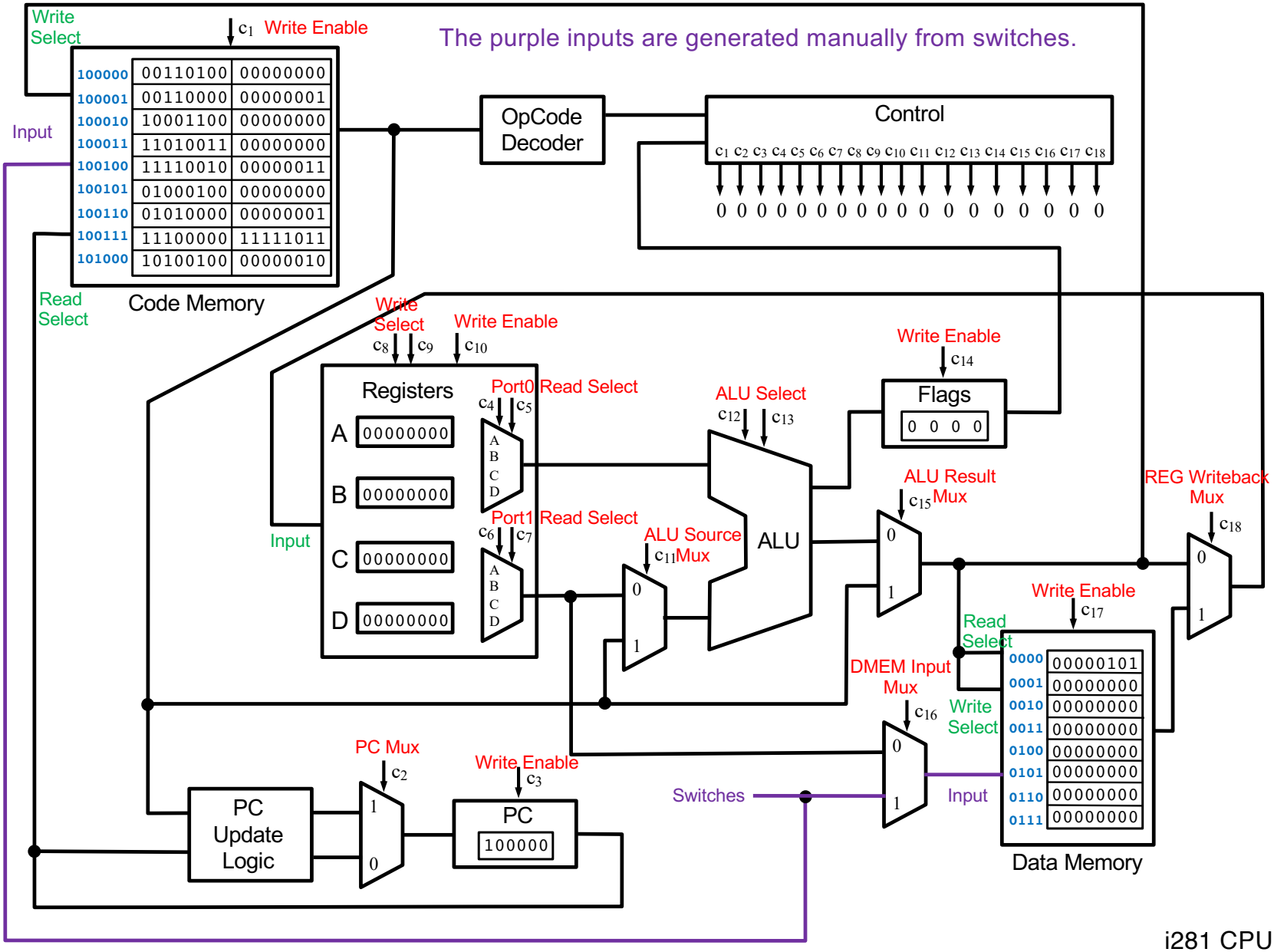

Control Signal #18
c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c14
Flags
0 0 0 0
4
c12 c13
ALU
c15
c11
c16
c17
c18
6 low
4 low
8 low
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
PC Update Logic
c2
c3
PC
100000
Switches
16
i281 CPU

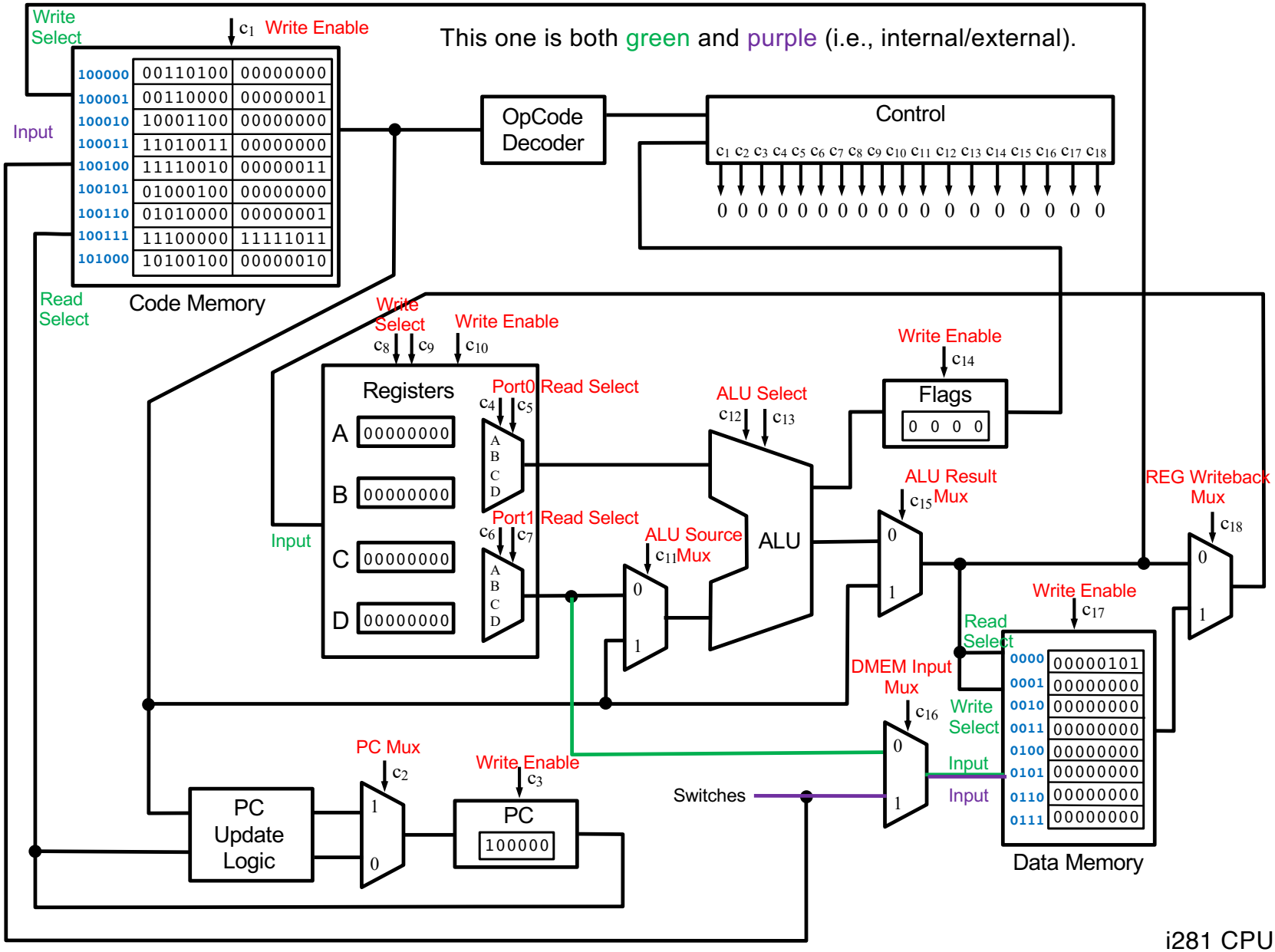

Control Signal #18
c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c14
Flags
0 0 0 0
4
c12 c13
ALU
c15
c11
c16
c17
c18
6 low
4 low
8 low
Switches
16
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
c2
c3
PC Update Logic
PC
100000
i281 CPU

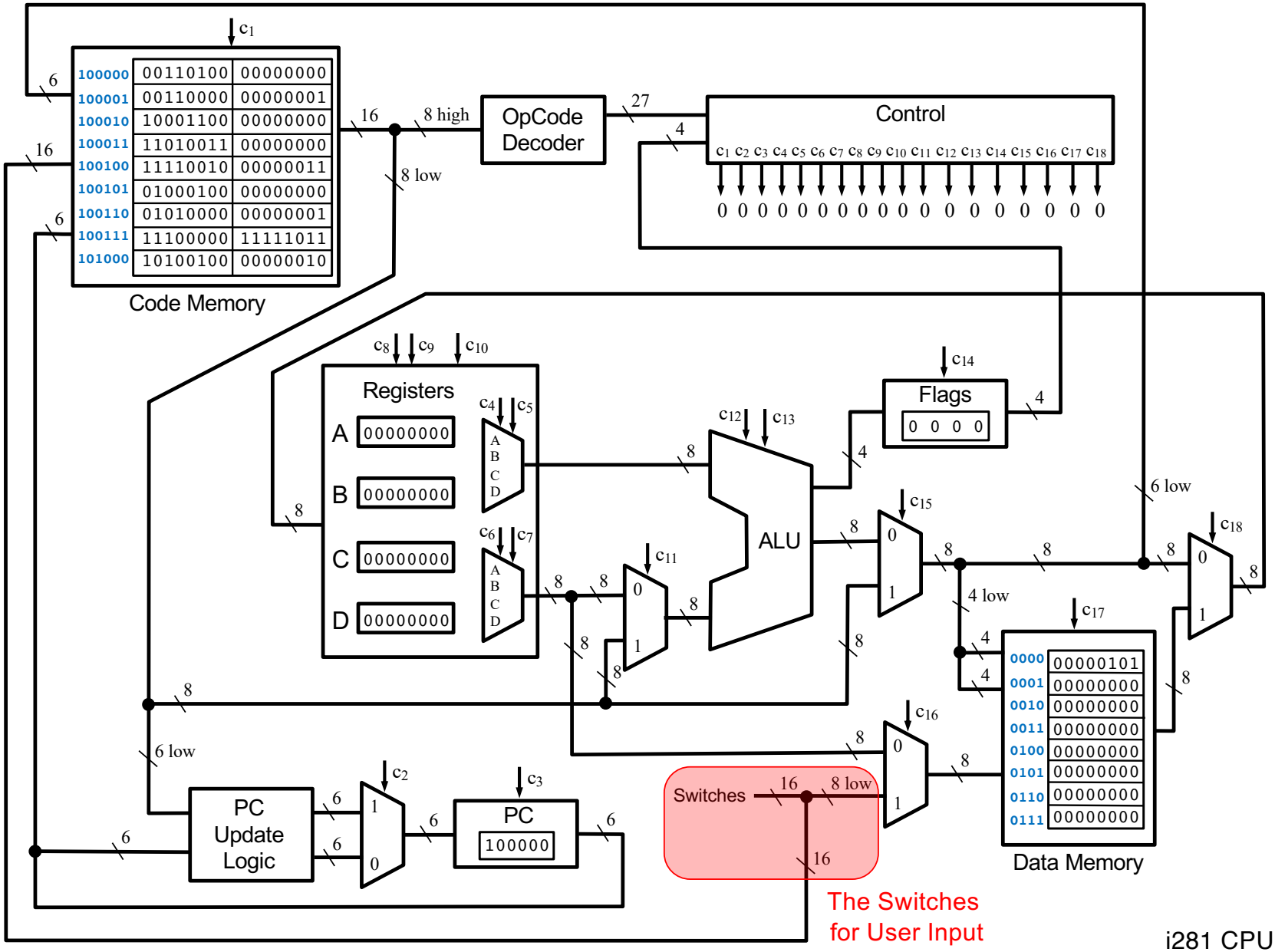

All Control Signals
c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9
c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c12 c13
ALU
c14
Flags
0 0 0 0
4
c15
c11
c18
6 low
4 low
c17
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
c16
Switches
8 low
c2
c3
PC Update Logic
PC
100000
i281 CPU

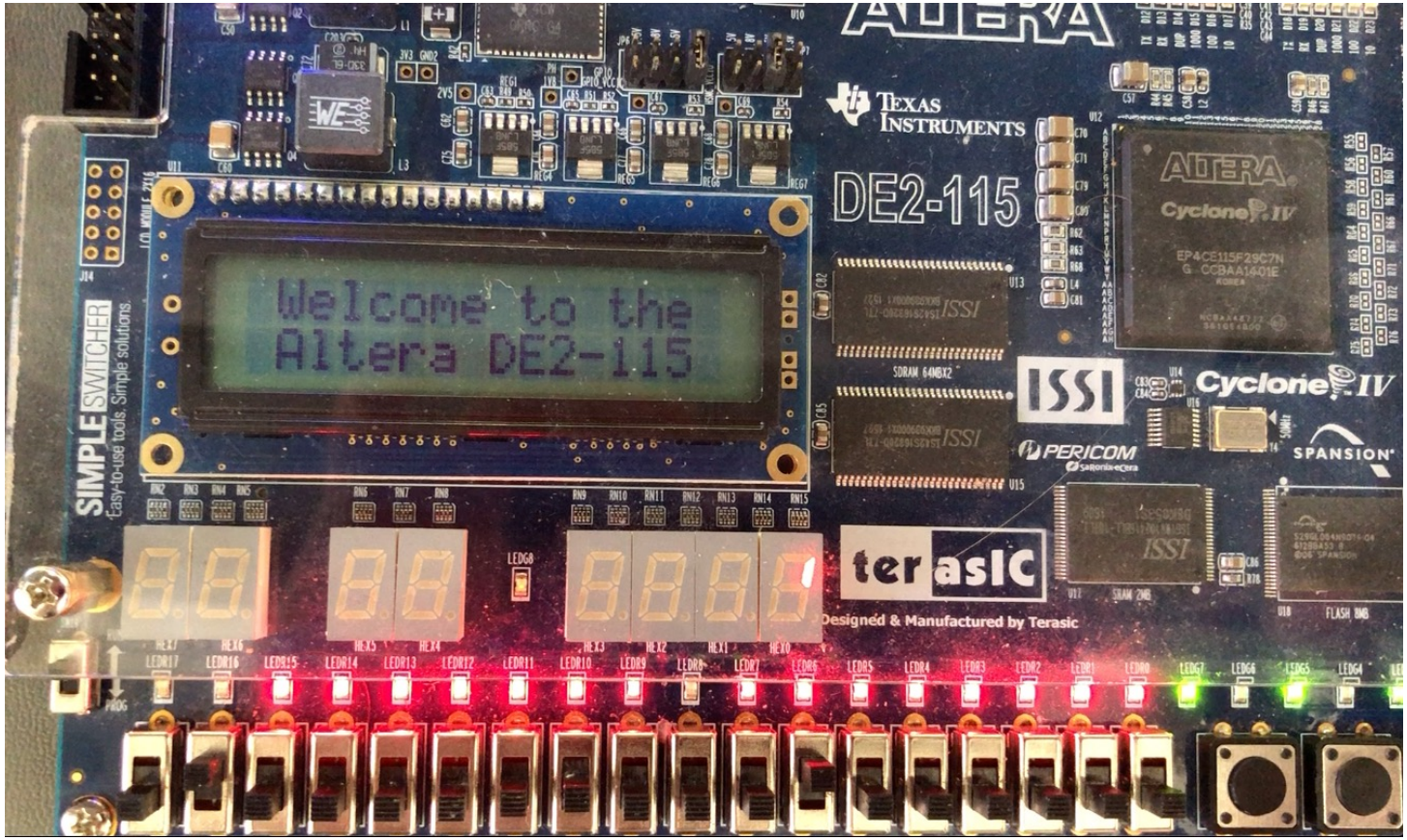

c1 Write Enable
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
OpCode Decoder
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Write Select
c8 c9
Write Enable
c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
Port0 Read Select
c4 c5
Port1 Read Select
c6 c7
A B C D
ALU Select
c12 c13
ALU
ALU Source
c11 Mux
Write Enable
c14
Flags
0 0 0 0
ALU Result
c15 Mux
REG Writeback
Mux
c18
Write Enable
c17
DMEM Input
Mux
c16
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
PC Mux
c2
Write Enable
c3
PC
100000
PC
Update
Logic
Switches
i281 CPU

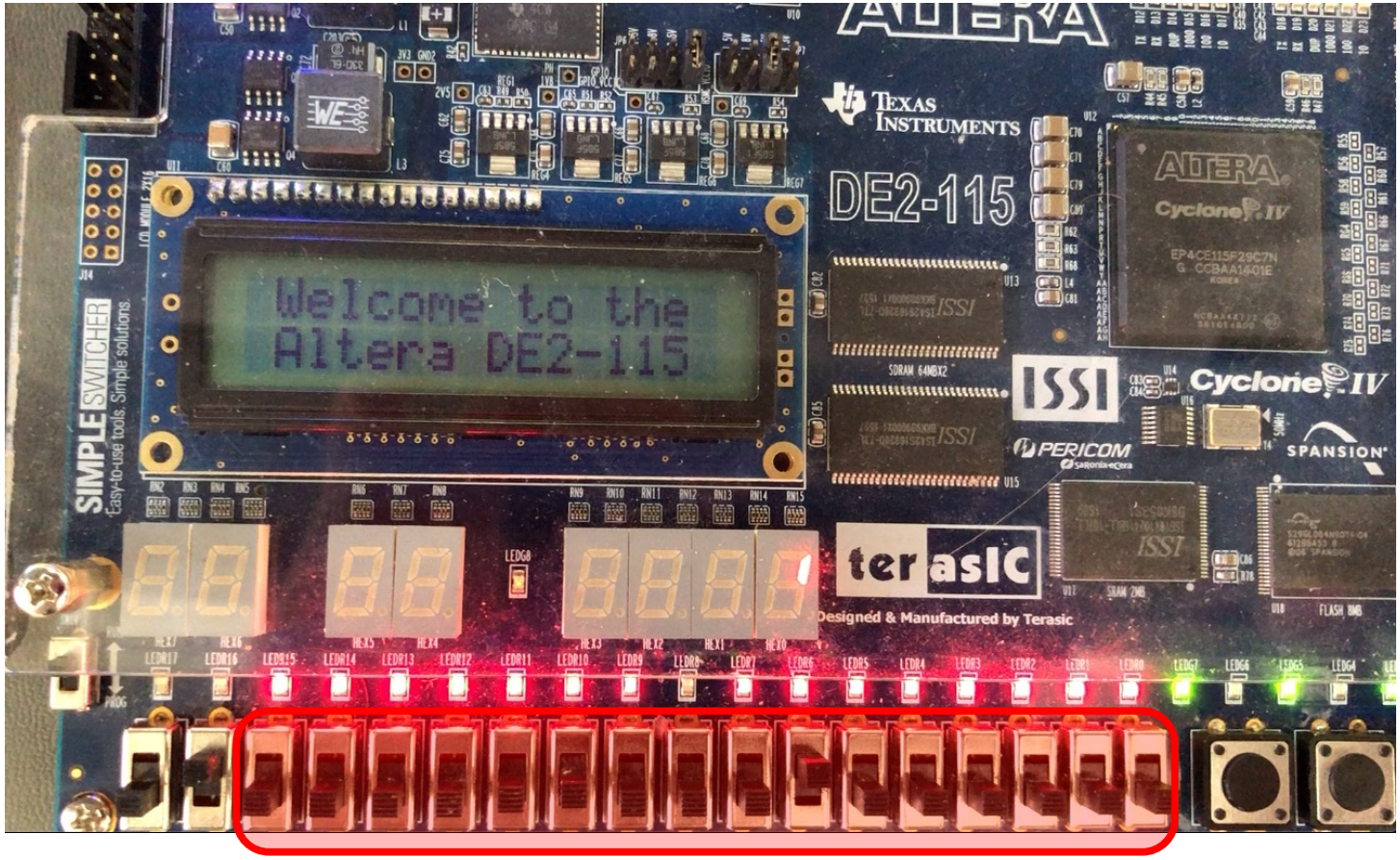

The green inputs are generated/computed internally.
Write Select
c1 Write Enable
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
Read Select
OpCode Decoder
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Write Select
c8 c9
Write Enable
c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
Input
Port0 Read Select
c4 c5
Port1 Read Select
c6 c7
A B C D
ALU Select
c12 c13
ALU
ALU Source
c11 Mux
Write Enable
c14
Flags
0 0 0 0
ALU Result
c15 Mux
REG Writeback
Mux
c18
Write Enable
c17
Read Select
Write Select
Input
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
DMEM Input
Mux
c16
Switches
PC Mux
c2
Write Enable
c3
PC
100000
PC Update Logic
i281 CPU


The purple inputs are generated manually from switches.
Write Select
c1 Write Enable
Input
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
Read Select
OpCode Decoder
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Write Select
c8 c9
Write Enable
c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
Port0 Read Select
c4 c5
Port1 Read Select
c6 c7
A B C D
Input
ALU Select
c12 c13
ALU
ALU Source
c11 Mux
Write Enable
c14
Flags
0 0 0 0
ALU Result
c15 Mux
REG Writeback
Mux
c18
Write Enable
c17
Read Select
DMEM Input
Mux
c16
Write Select
Switches
Input
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
PC Mux
c2
Write Enable
c3
PC Update Logic
PC
100000
i281 CPU

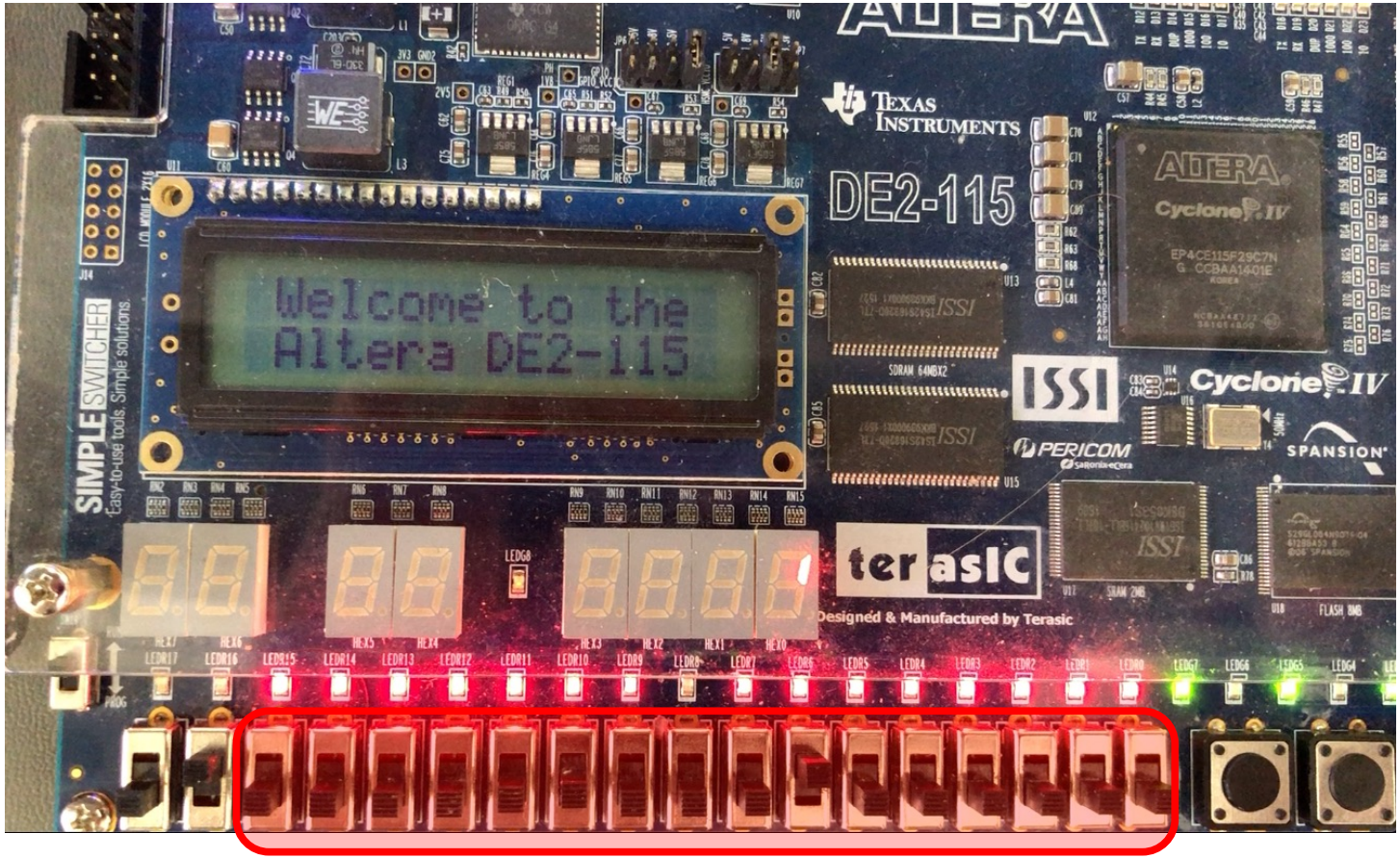

This one is both green and purple (i.e., internal/external).
Write Select
c1 Write Enable
Input
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
Read Select
OpCode Decoder
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Write Select
c8 c9
Write Enable
c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
Port0 Read Select
c4 c5
Port1 Read Select
c6 c7
A B C D
Input
ALU Select
c12 c13
ALU
ALU Source
c11 Mux
Write Enable
c14
Flags
0 0 0 0
ALU Result
c15 Mux
REG Writeback
Mux
c18
Write Enable
c17
Read Select
DMEM Input
Mux
c16
Write Select
Input
Input
Switches
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
PC Mux
c2
Write Enable
c3
PC Update Logic
PC
100000
i281 CPU


c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c12 c13
ALU
c14
Flags
0 0 0 0
4
c15
c11
c16
c17
c18
6 low
4 low
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
PC Update Logic
c2
c3
PC
100000
6 low
Switches
16
8 low
The Switches for User Input
i281 CPU

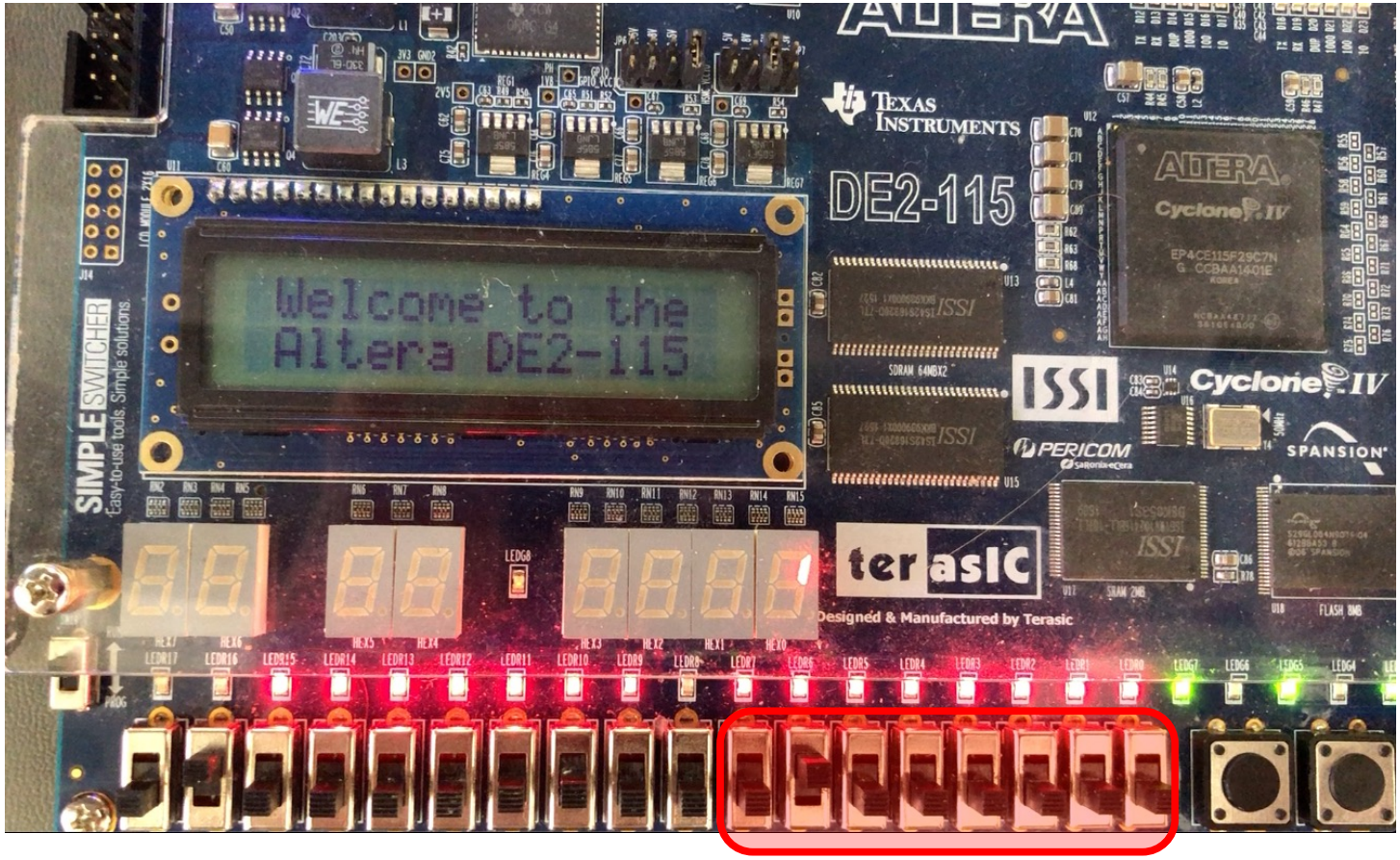

Welcome to the
Altera DE2-115
DE2-115
ALTERA
Cyclone IV
EP4CE115F29C7N
TEXAS INSTRUMENTS
ISSI
PERICOM
SPANSION
SDRAM 64MBX2
SRAM 2MB
FLASH 8MB
SIMPLE SWITCHER
Easy-to-use tools. Simple solutions.
terasIC
Designed & Manufactured by Terasic



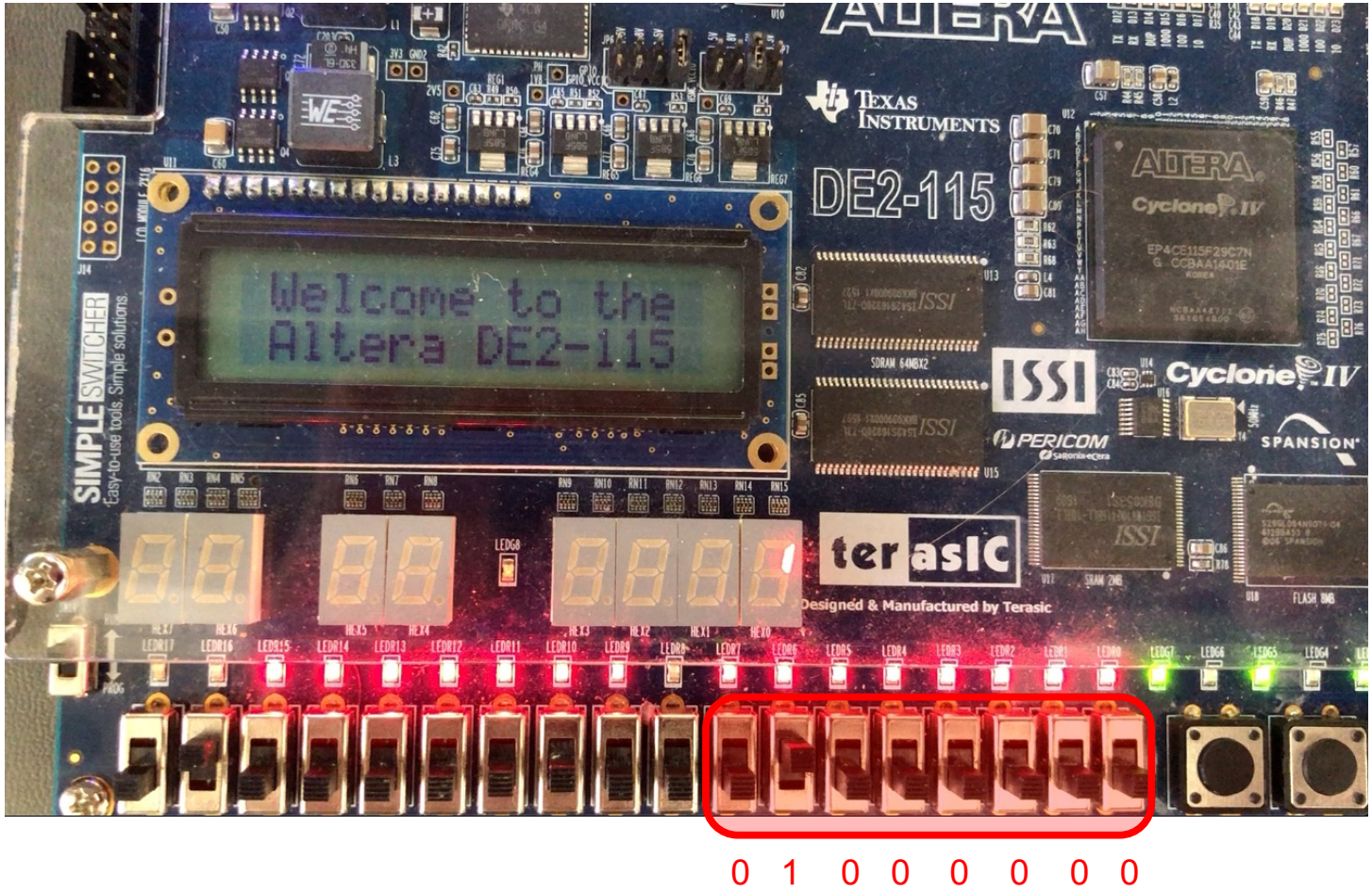
These 16 switches are used for input into the Code Memory.

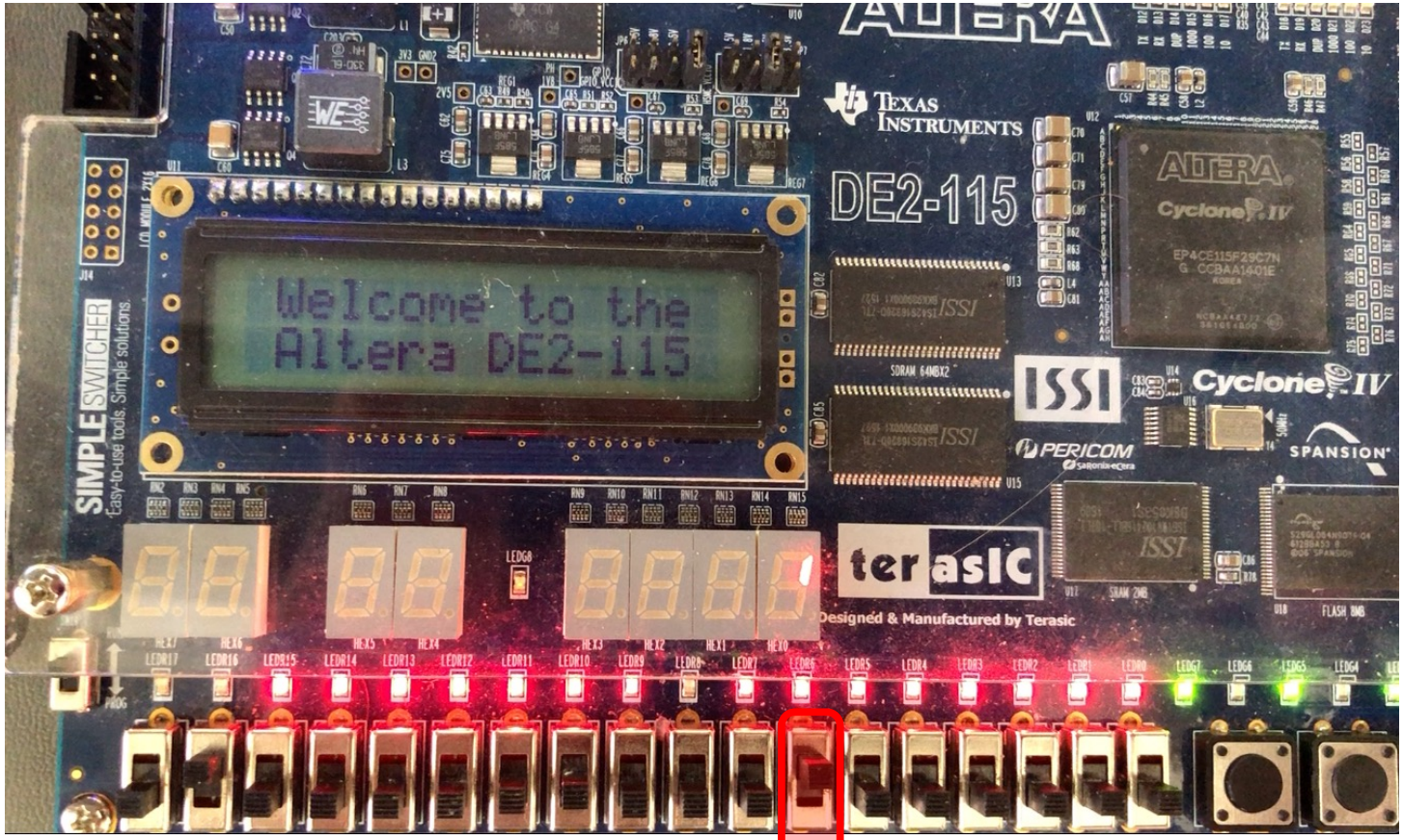


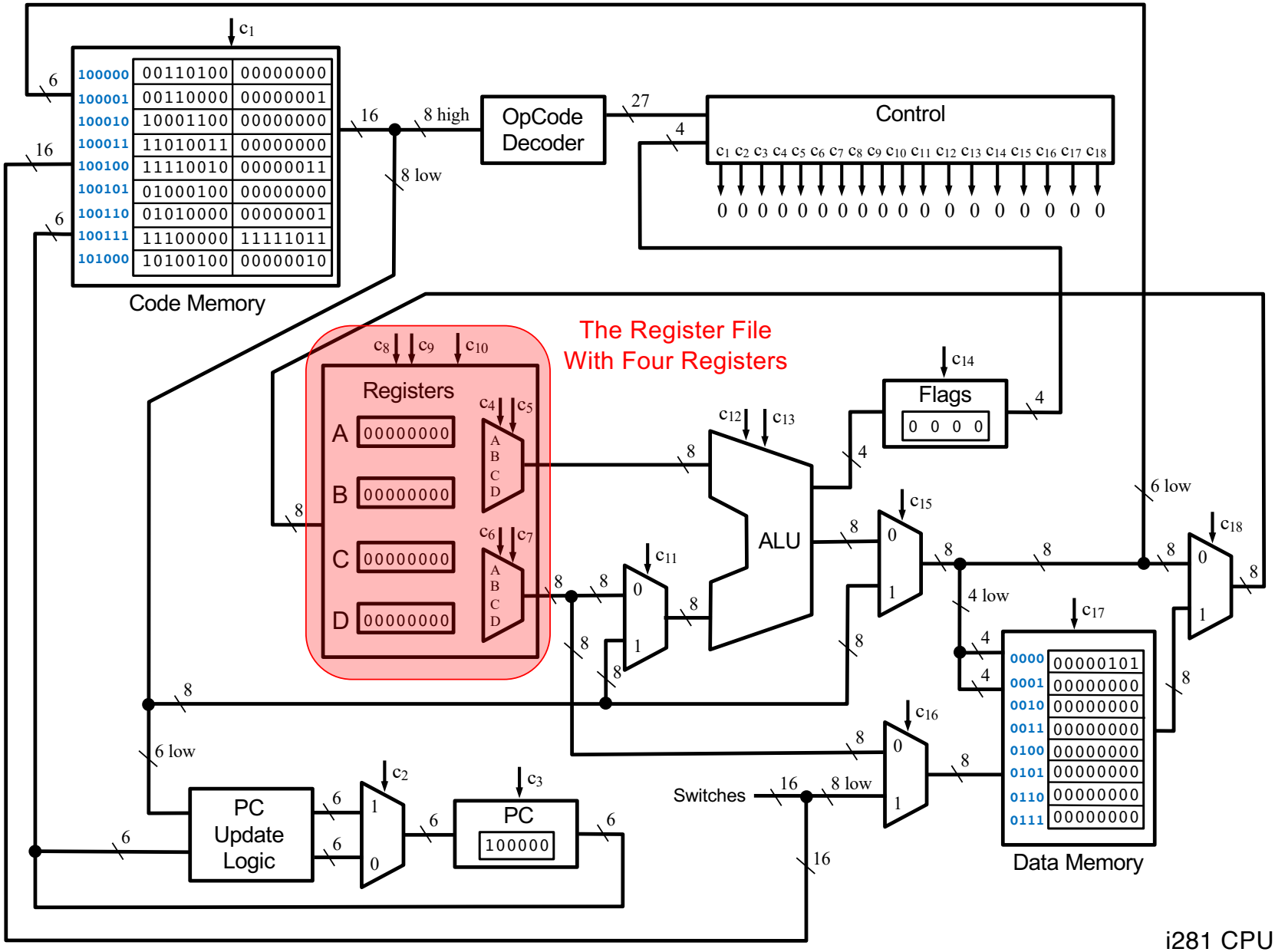
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0

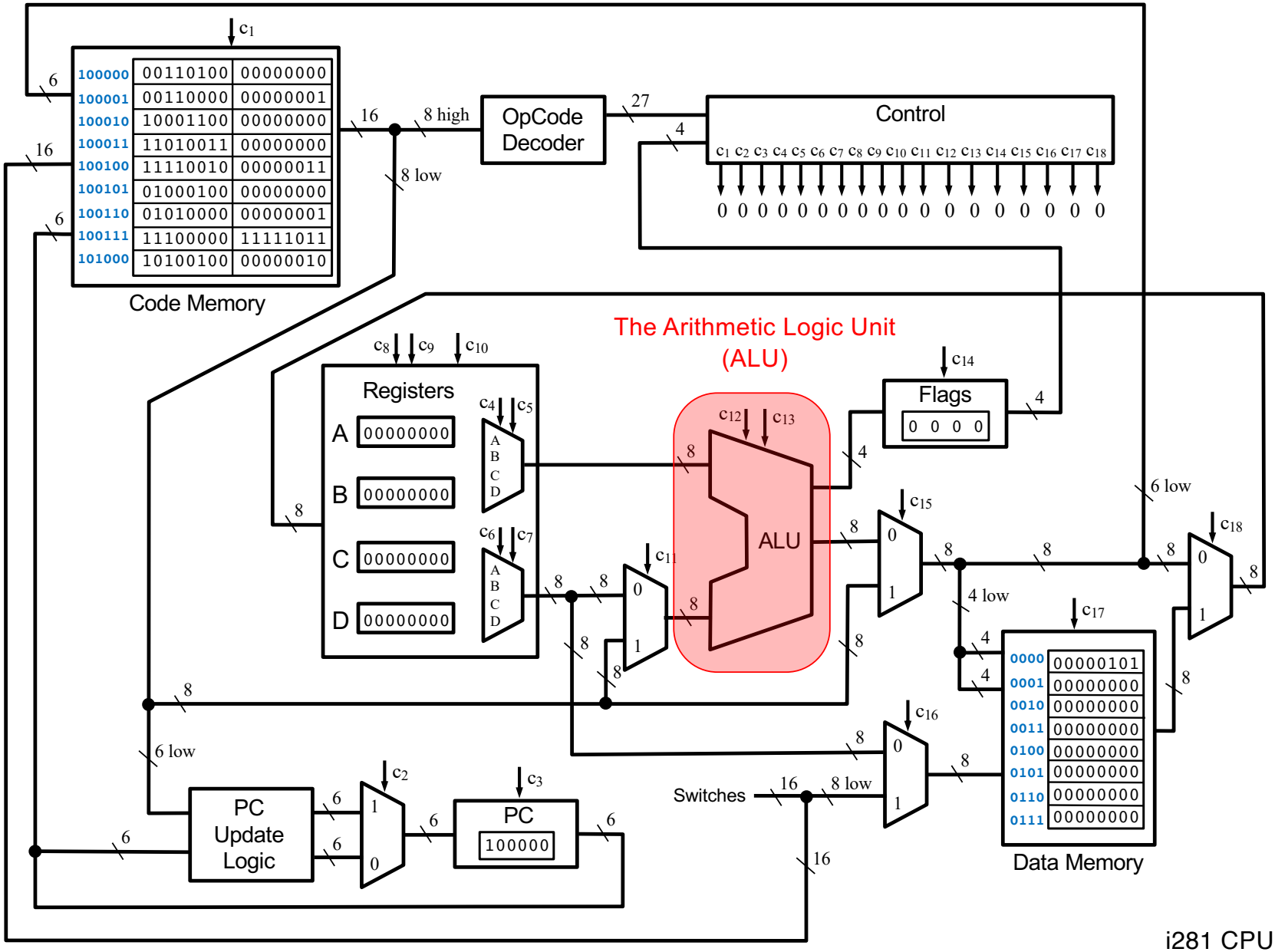


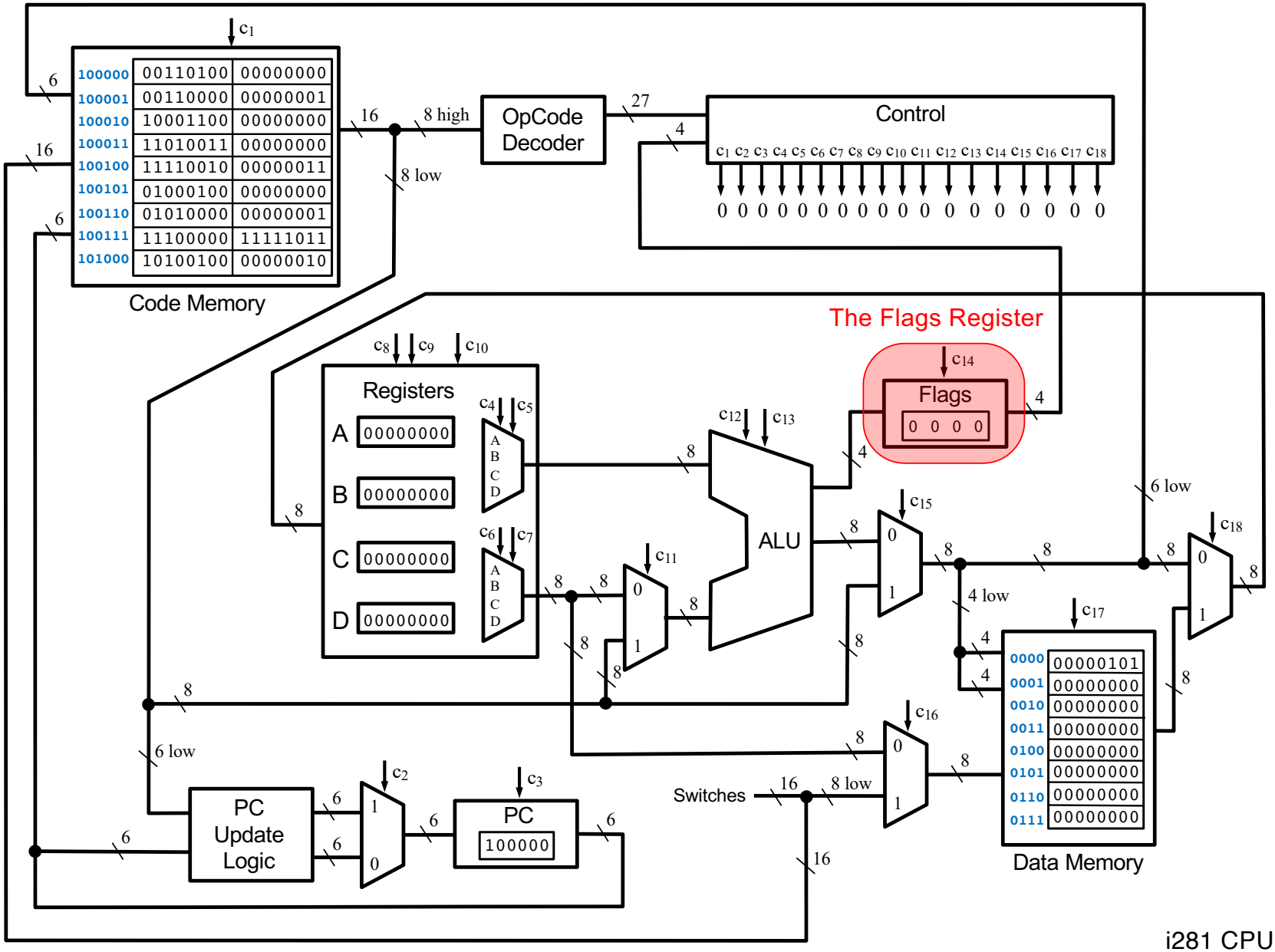
These 8 switches are used for input into the Data Memory.

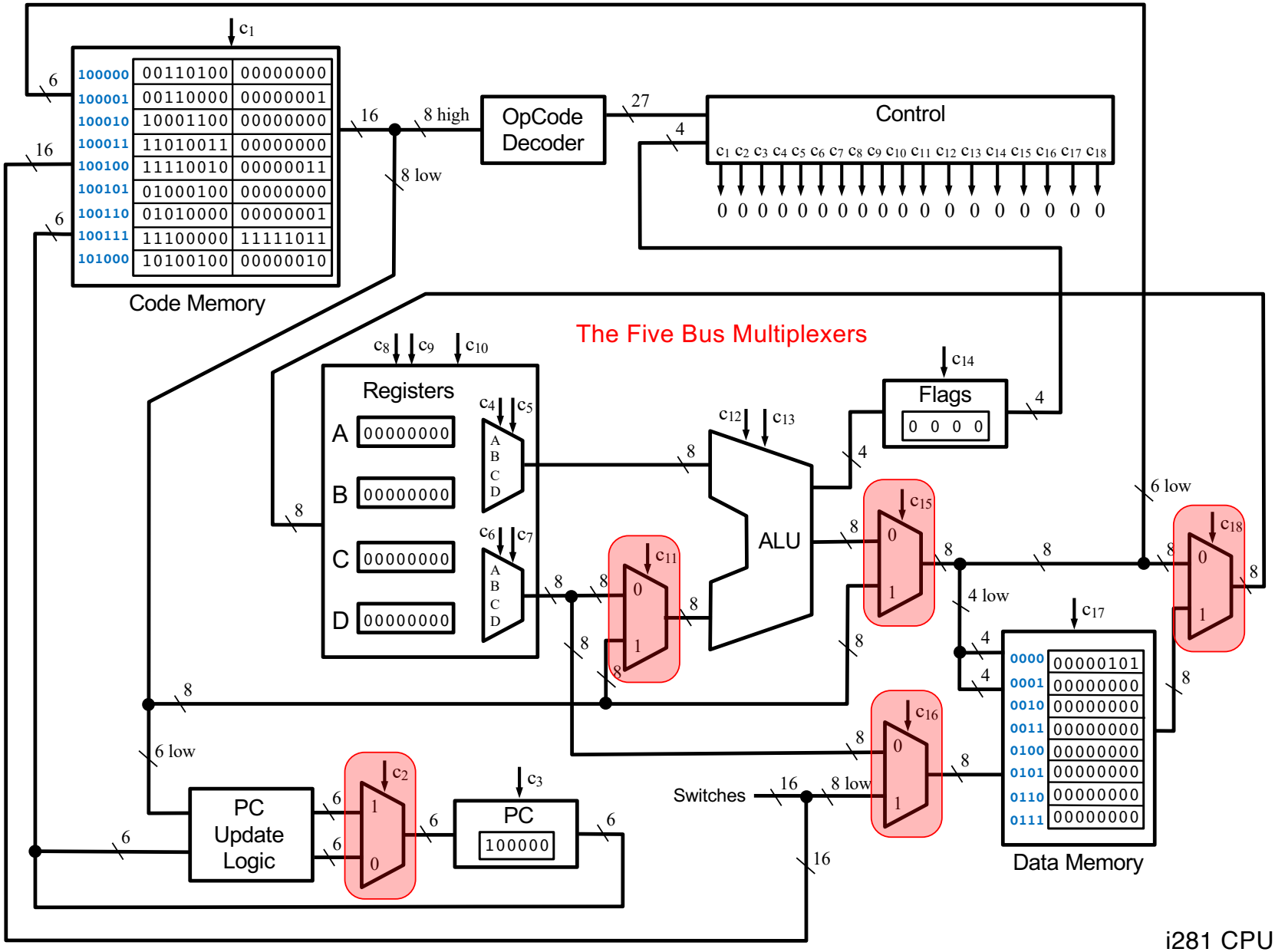

Welcome to the
Altera DE2-115
DE2-115
ALTERA
Cyclone IV
EP4CE115F29C7N
Texas Instruments
ISSI
PERICOM
SPANSION
terasIC
Designed & Manufactured by Terasic
SIMPLE SWITCHER
Easy-to-use tools. Simple solutions.
0 1 0 0 0 0 0 0

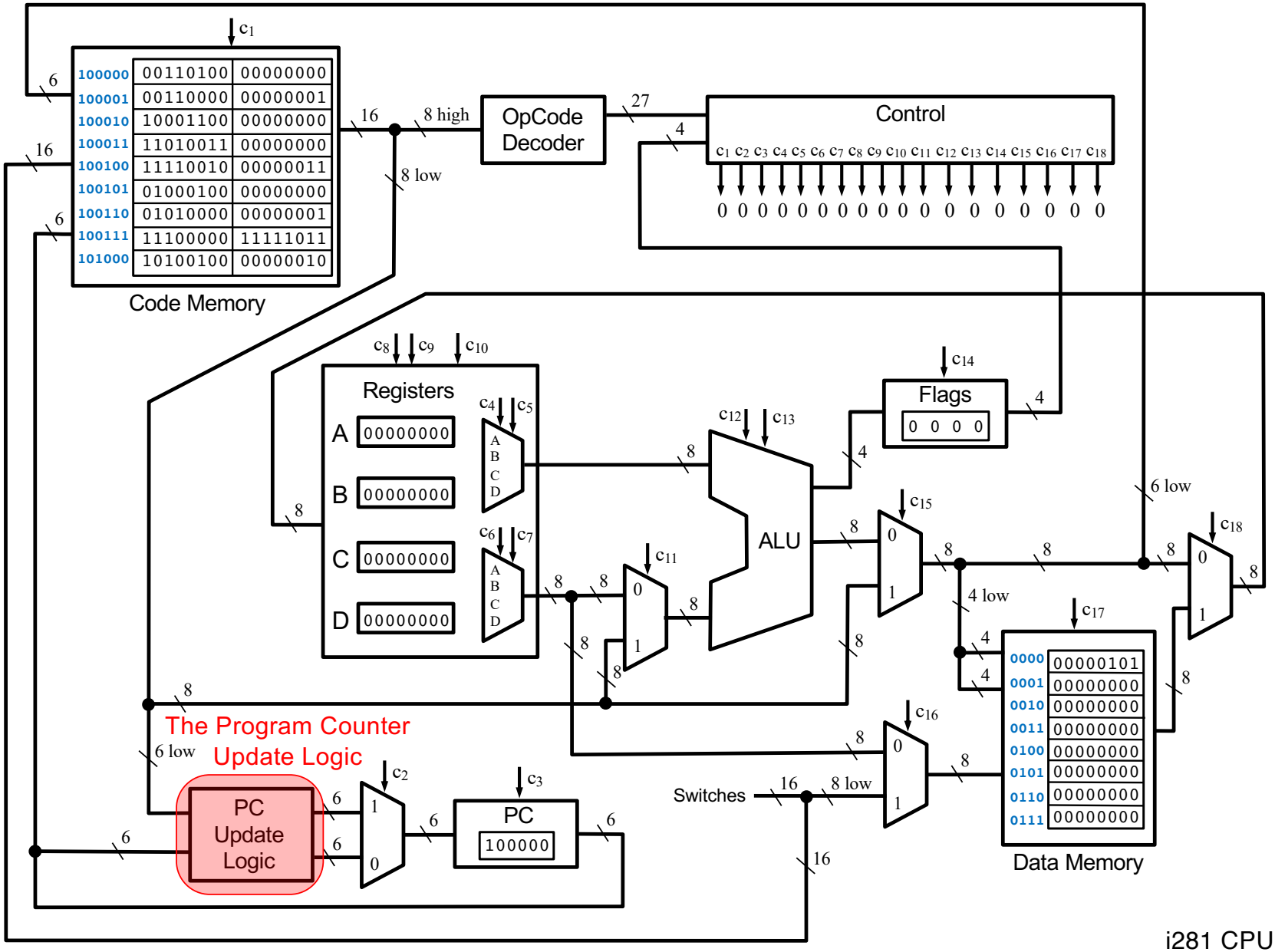

Welcome to the
Altera DE2-115
DE2-115
Cyclone IV
terasIC
Designed & Manufactured by Terasic
0 1 0 0 0 0 0 0
This switch controls
the paddle in PONG

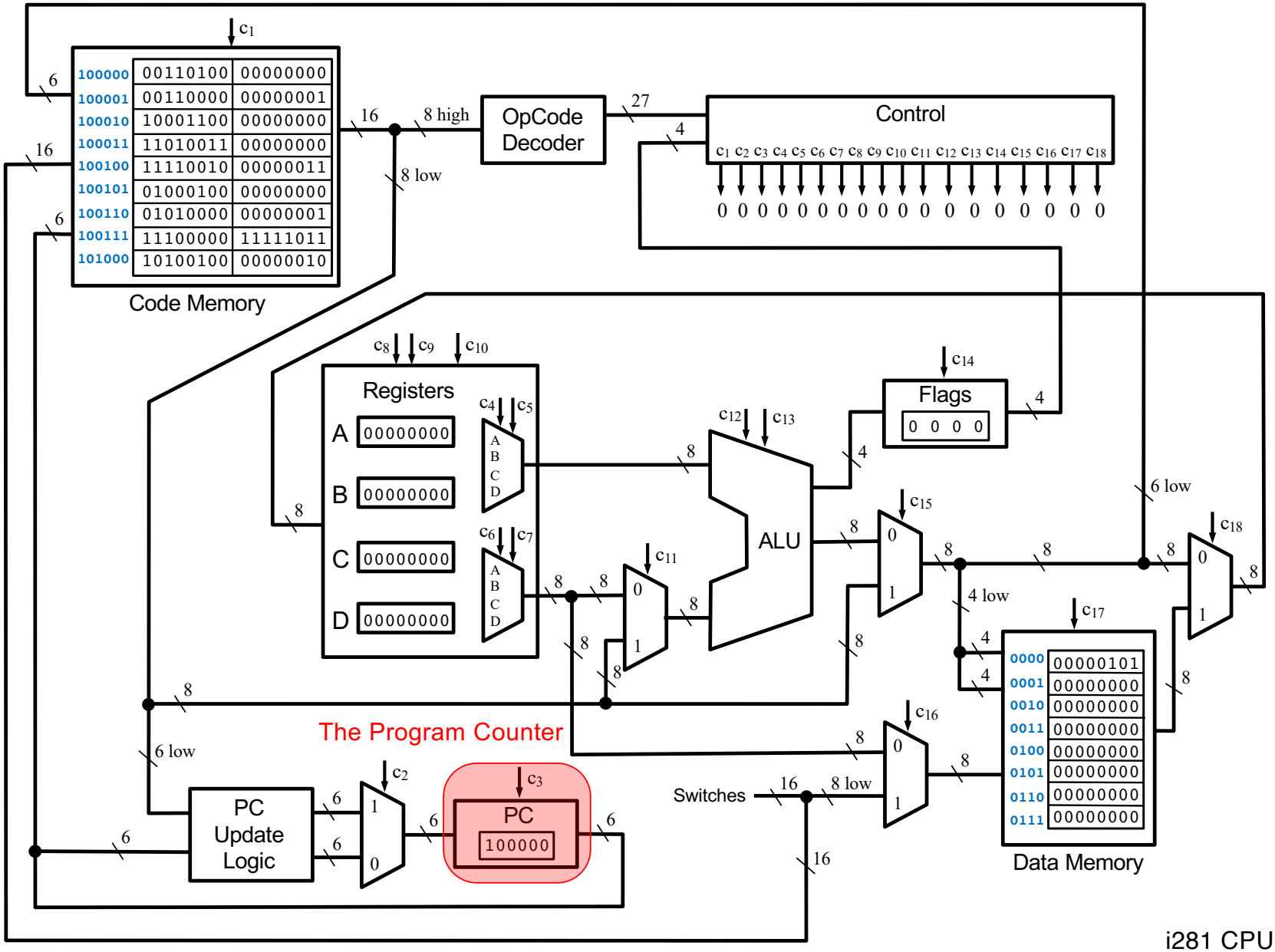

c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
The Register File
With Four Registers
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
c14
Flags
0 0 0 0
c12 c13
ALU
c15
c11
c16
c17
c18
6 low
4 low
8 low
Switches
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
PC Update Logic
c2
c3
PC
100000
i281 CPU

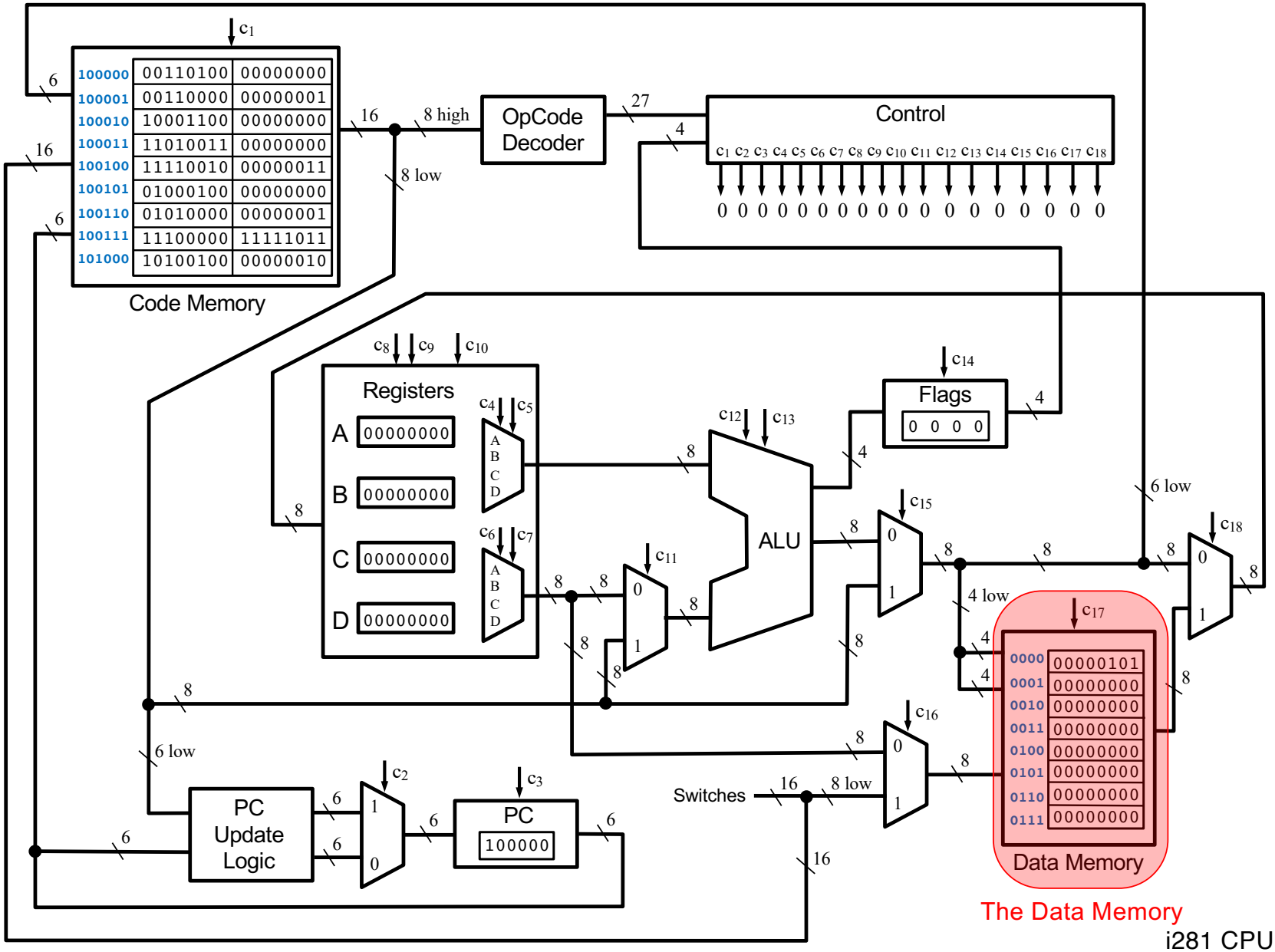

c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
The Arithmetic Logic Unit (ALU)
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
c12 c13
ALU
c14
Flags
0 0 0 0
c15
c11
c16
c17
c18
6 low
4 low
8 low
Switches
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
c2
c3
PC Update Logic
PC
100000
i281 CPU

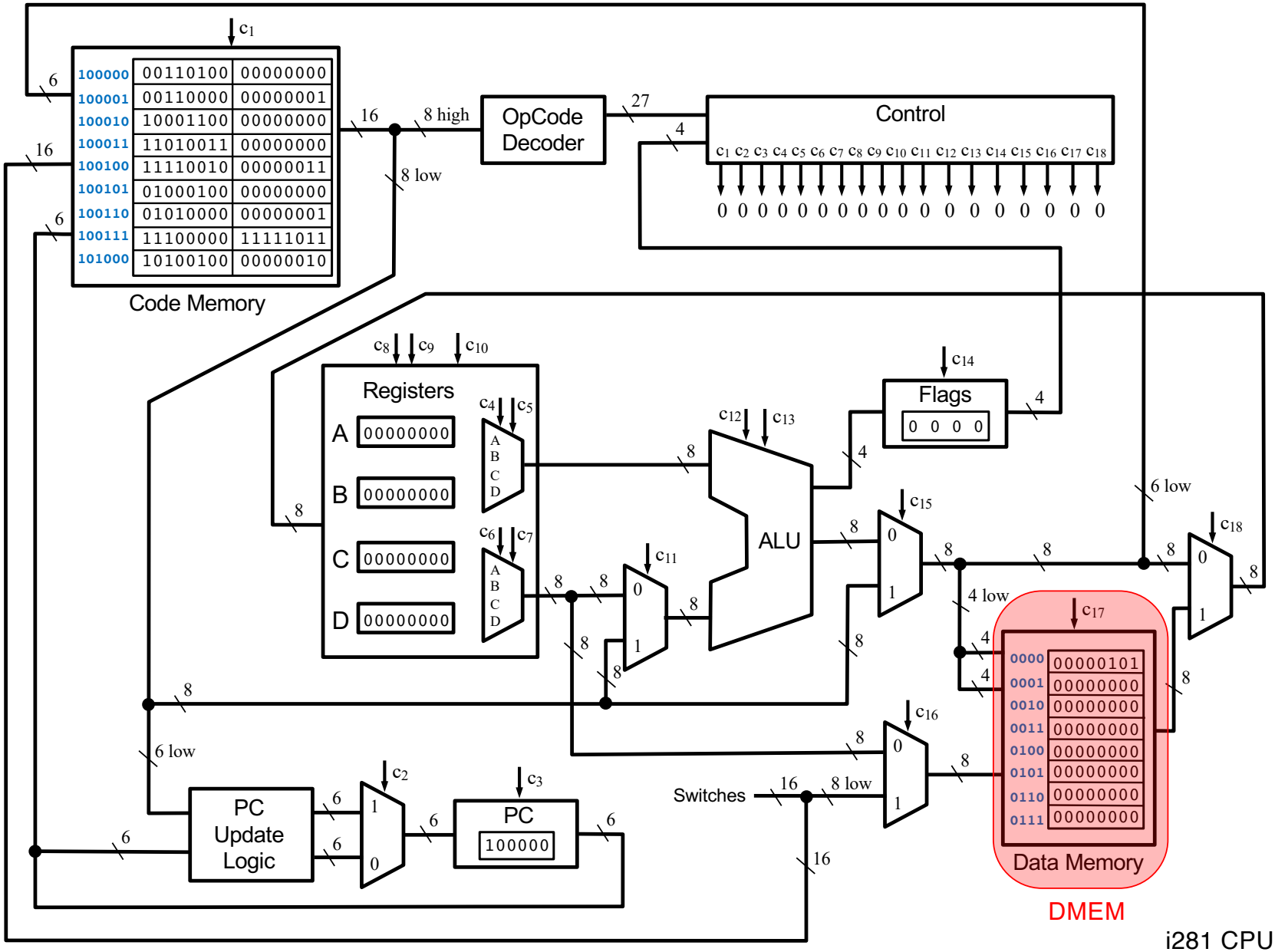

c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
The Flags Register
c14
Flags
0 0 0 0
4
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
c12 c13
ALU
c15
c11
c16
c17
c18
6 low
4 low
8 low
8 high
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
Switches
PC Update Logic
c2
c3
PC
100000
i281 CPU


c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
The Five Bus Multiplexers
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c12 c13
ALU
c14
Flags
0 0 0 0
4
c15
c11
c16
c18
6 low
4 low
c17
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
Switches
16
8 low
6 low
PC Update Logic
c2
c3
PC
100000
i281 CPU


c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c14
Flags
0 0 0 0
4
c12 c13
ALU
c15
c11
c16
c17
c18
6 low
4 low
8 low
Switches
16
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
The Program Counter
Update Logic
6 low
PC
Update
Logic
c2
c3
PC
100000
i281 CPU


c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c14
Flags
0 0 0 0
4
c12 c13
ALU
c15
c11
c16
c17
c18
6 low
4 low
8 low
Switches
16
The Program Counter
c2
c3
PC
100000
PC Update Logic
6 low
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
i281 CPU

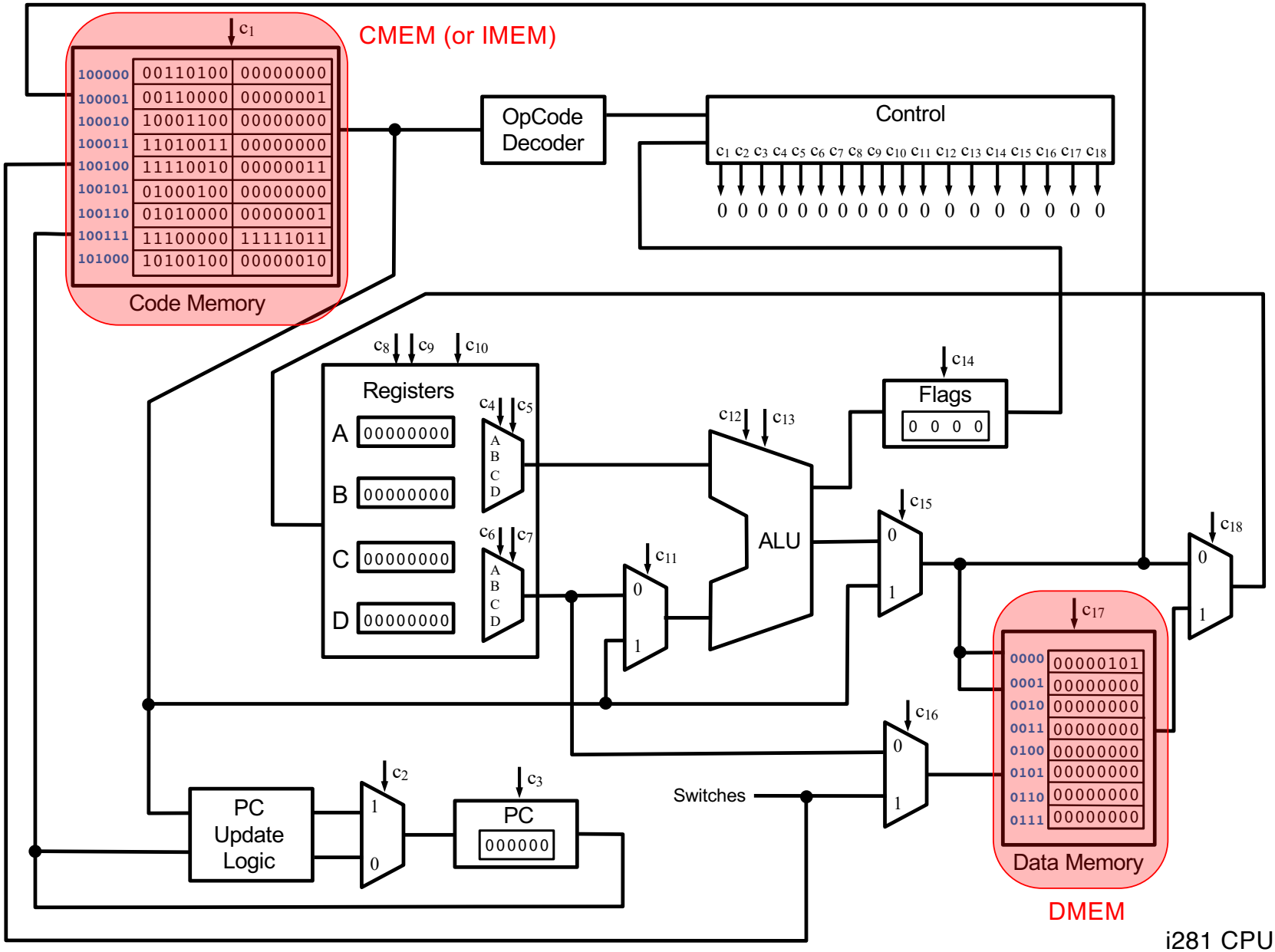

c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c12 c13
ALU
c14
Flags
0 0 0 0
4
c15
c11
c16
c17
c18
6 low
4 low
8 low
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
PC Update Logic
c2
c3
PC
100000
Switches
16
The Data Memory
i281 CPU


c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c14
Flags
0 0 0 0
4
c12 c13
ALU
c15
c11
c16
6 low
c18
4 low
c17
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
DMEM
PC Update Logic
c2
c3
PC
100000
Switches
16
8 low
i281 CPU

The Memory Layout

Memory Layout

- The i281 CPU uses two different memories


CMEM (or IMEM)
c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
OpCode Decoder
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
c11
c12 c13
ALU
c14
Flags
0 0 0 0
c15
c16
c17
c18
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
DMEM
Switches
c2
c3
PC Update Logic
PC
000000
i281 CPU

Memory Layout

- The i281 CPU uses two different memories

- Data Memory

  16 x 8 bits

- Code Memory

  64 x 16 bits

Memory Layout

- The i281 CPU uses two different memories

- Data Memory

  16 x 8 bits

- Code Memory

  64 x 16 bits

Note that they have different number of bits

Memory Layout

- The i281 CPU uses two different memories

- Data Memory

  16 x 8 bits (only 16 bytes!)

- Code Memory

  64 x 16 bits (only 128 bytes!)

This is a combined total of 144 bytes!

Memory Layout

- The i281 CPU uses two different memories

- Data Memory

  16 x 8 bits (only 16 bytes!)

- Code Memory

  64 x 16 bits (only 128 bytes!)

This is a combined total of 144 bytes!

Which is enough to represent 48 pixels on this slide!

Data Memory Layout

- **Organized as one contiguous block**

- **Data Memory**
  - **Random access**
  - **Read/write memory**
  - **16 x 8 bits**
  - **Only 16 bytes!**

Data Memory Layout

- **Organized as one contiguous block**

- **Data Memory**
  - **Random access**
  - **Read/write memory**
  - **16 x 8 bits**
  - **Only 16 bytes!**

Implemented as a register file with 16 registers, each of which is 8-bits wide.

The register file has one read port and one write port. It also has a write enable input.

Video Card

Memory-mapped video memory

The first 8 bytes of the data memory are connected to the 7-segment displays on the Altera board

Writing to these memory cells automatically lights up the displays, which use only the 4 least significant bits

In video game mode each LED is controlled separately using a different set of 7-segment decoders & all 8 bits

The contents of the second 8 bytes of the data memory cannot be visualized, but programs can still use them

Data Memory Contents for the Bubble Sort Program

```
00000111
00000011
00000010
00000001
00000110
00000100
00000101
00001000
00000111
00000000
00000000
00000000
00000000
00000000
00000000
00000000
```

Data Memory Contents for the Bubble Sort Program

Address	Data
0000	00000111
0001	00000011
0010	00000010
0011	00000001
0100	00000110
0101	00000100
0110	00000101
0111	00001000
1000	00000111
1001	00000000
1010	00000000
1011	00000000
1100	00000000
1101	00000000
1110	00000000
1111	00000000

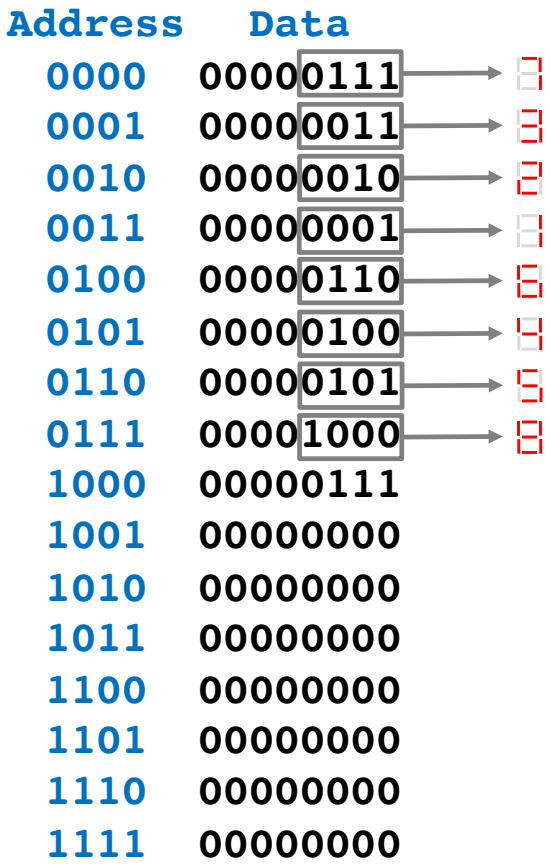
Data Memory Contents for the Bubble Sort Program

Address	Data	Comment
0000	00000111	//array[0]
0001	00000011	//array[1]
0010	00000010	//array[2]
0011	00000001	//array[3]
0100	00000110	//array[4]
0101	00000100	//array[5]
0110	00000101	//array[6]
0111	00001000	//array[7]
1000	00000111	//last
1001	00000000	//temp
1010	00000000	
1011	00000000	
1100	00000000	
1101	00000000	
1110	00000000	
1111	00000000	

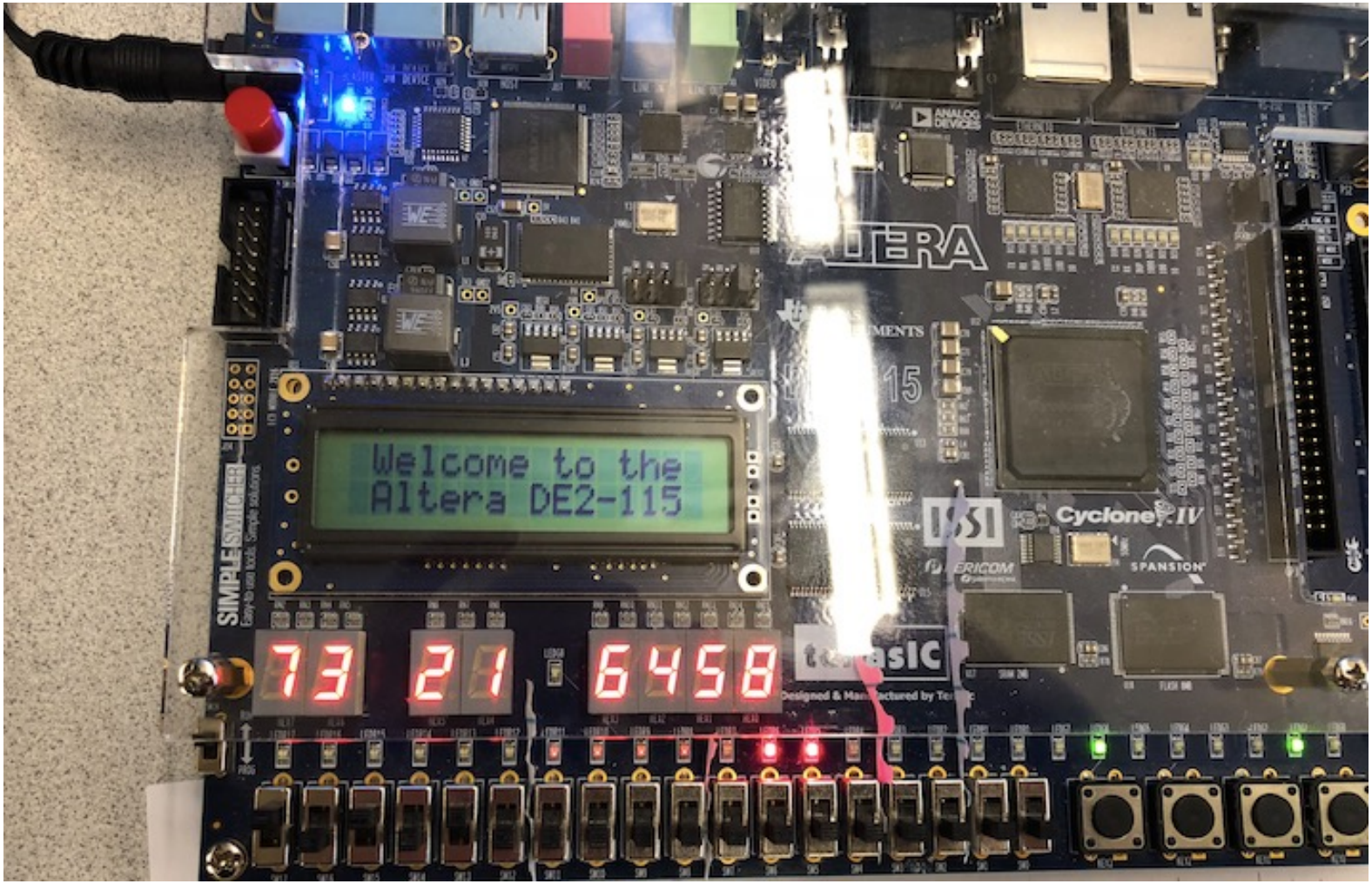
Memory-Mapped Video Memory

Address	Data
0000	00000111
0001	00000011
0010	00000010
0011	00000001
0100	00000110
0101	00000100
0110	00000101
0111	00001000
1000	00000111
1001	00000000
1010	00000000
1011	00000000
1100	00000000
1101	00000000
1110	00000000
1111	00000000

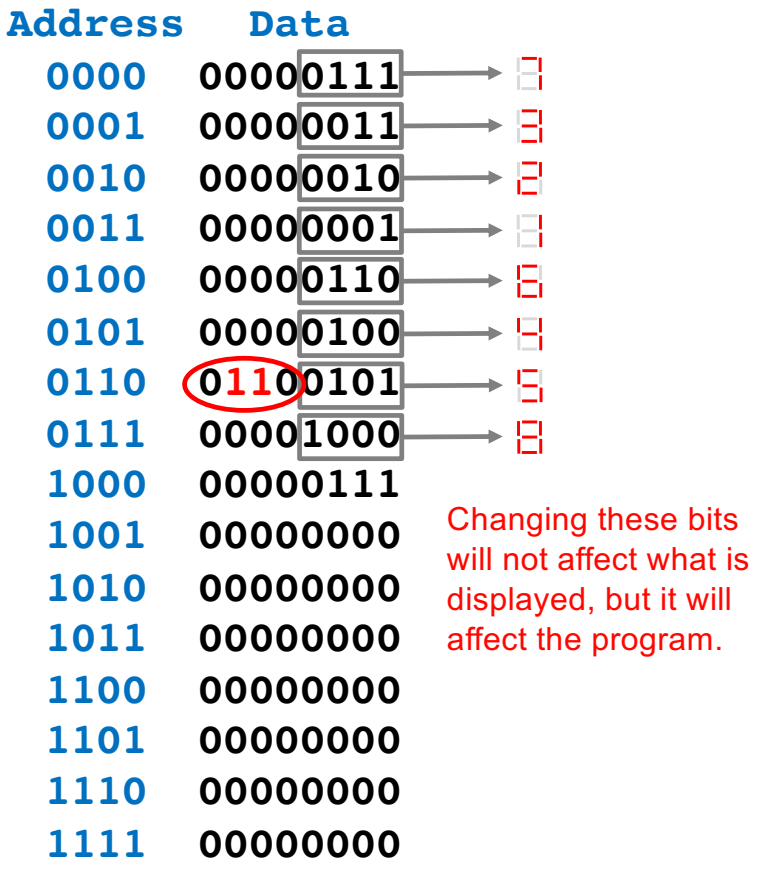
Memory-Mapped Video Memory



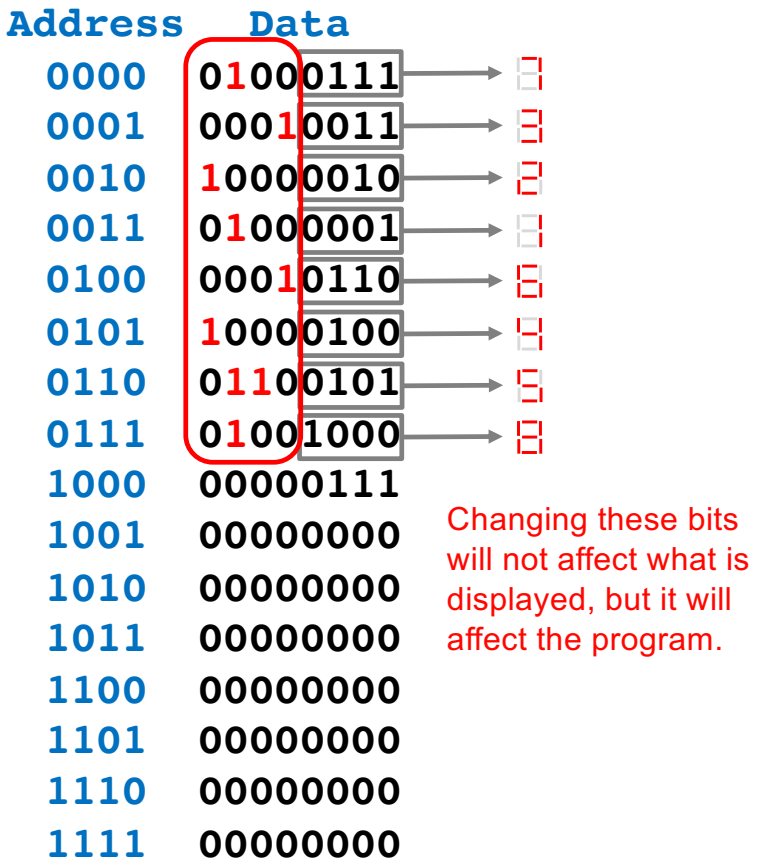
Memory-Mapped Video Memory



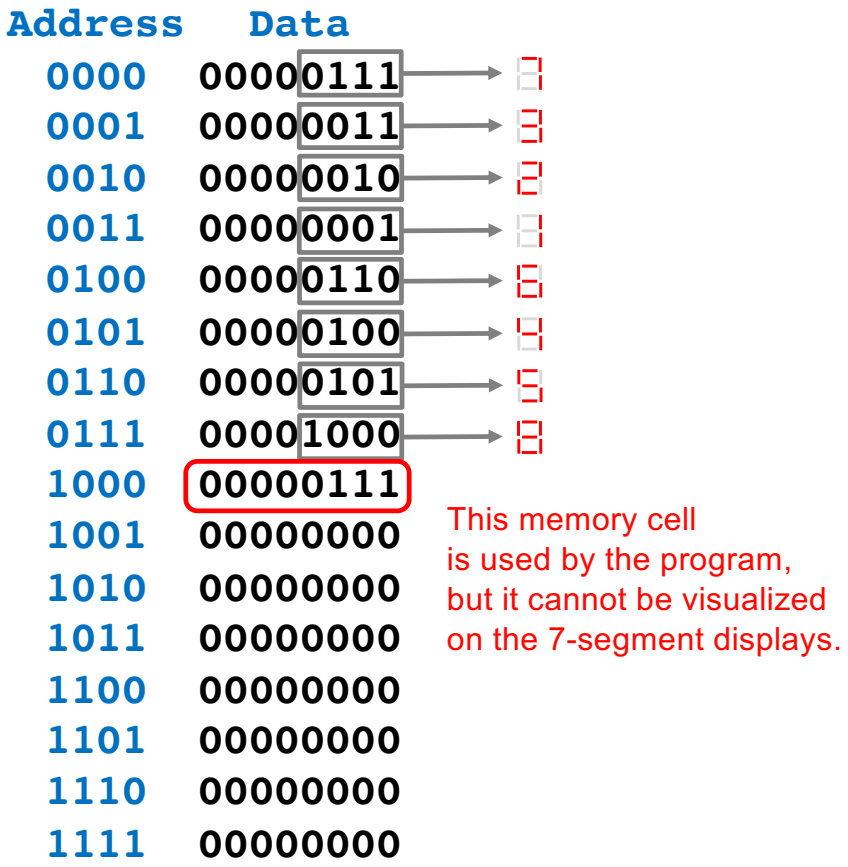
Memory-Mapped Video Memory



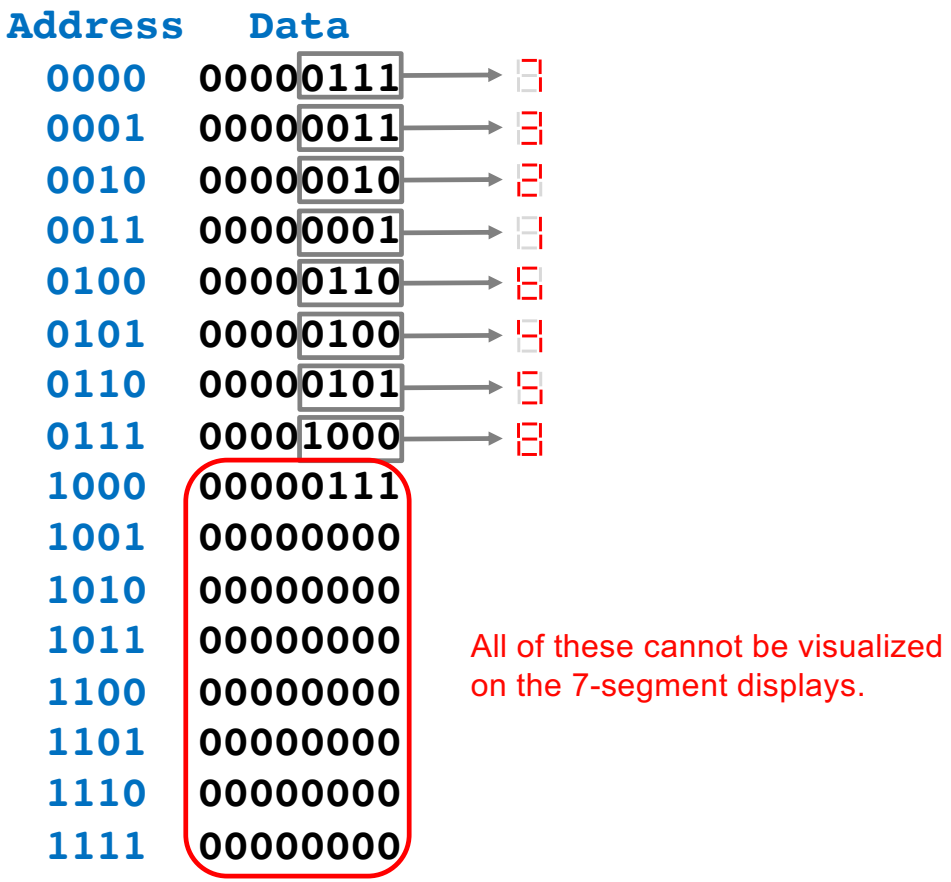
Memory-Mapped Video Memory



Memory-Mapped Video Memory



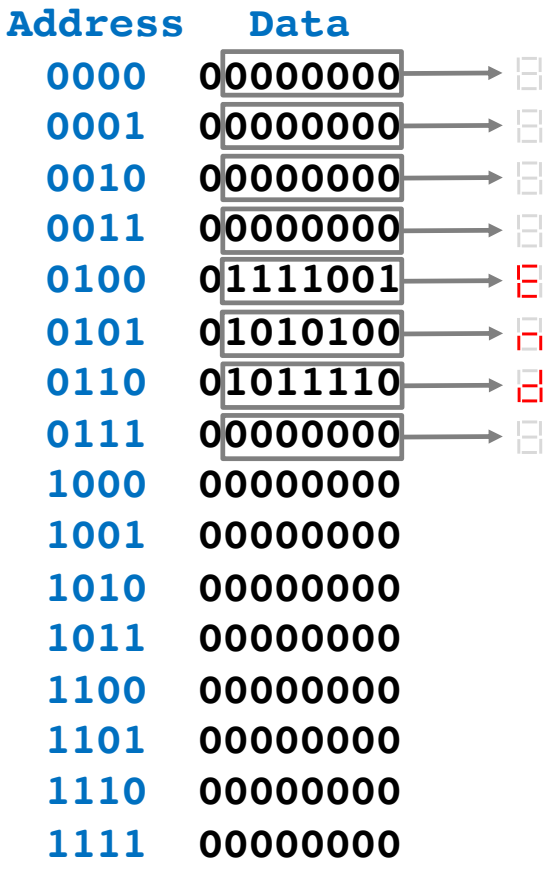
Memory-Mapped Video Memory



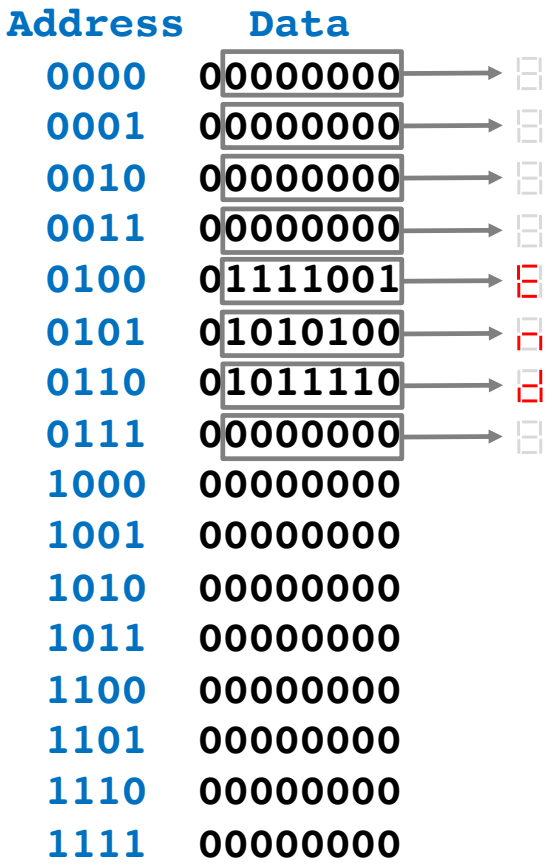
Memory-Mapped Video Memory in Video Game Mode

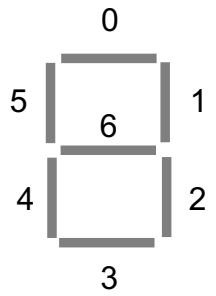
Address	Data
0000	00000000
0001	00000000
0010	00000000
0011	00000000
0100	01111001
0101	01010100
0110	01011110
0111	00000000
1000	00000000
1001	00000000
1010	00000000
1011	00000000
1100	00000000
1101	00000000
1110	00000000
1111	00000000

Memory-Mapped Video Memory in Video Game Mode



Memory-Mapped Video Memory in Video Game Mode





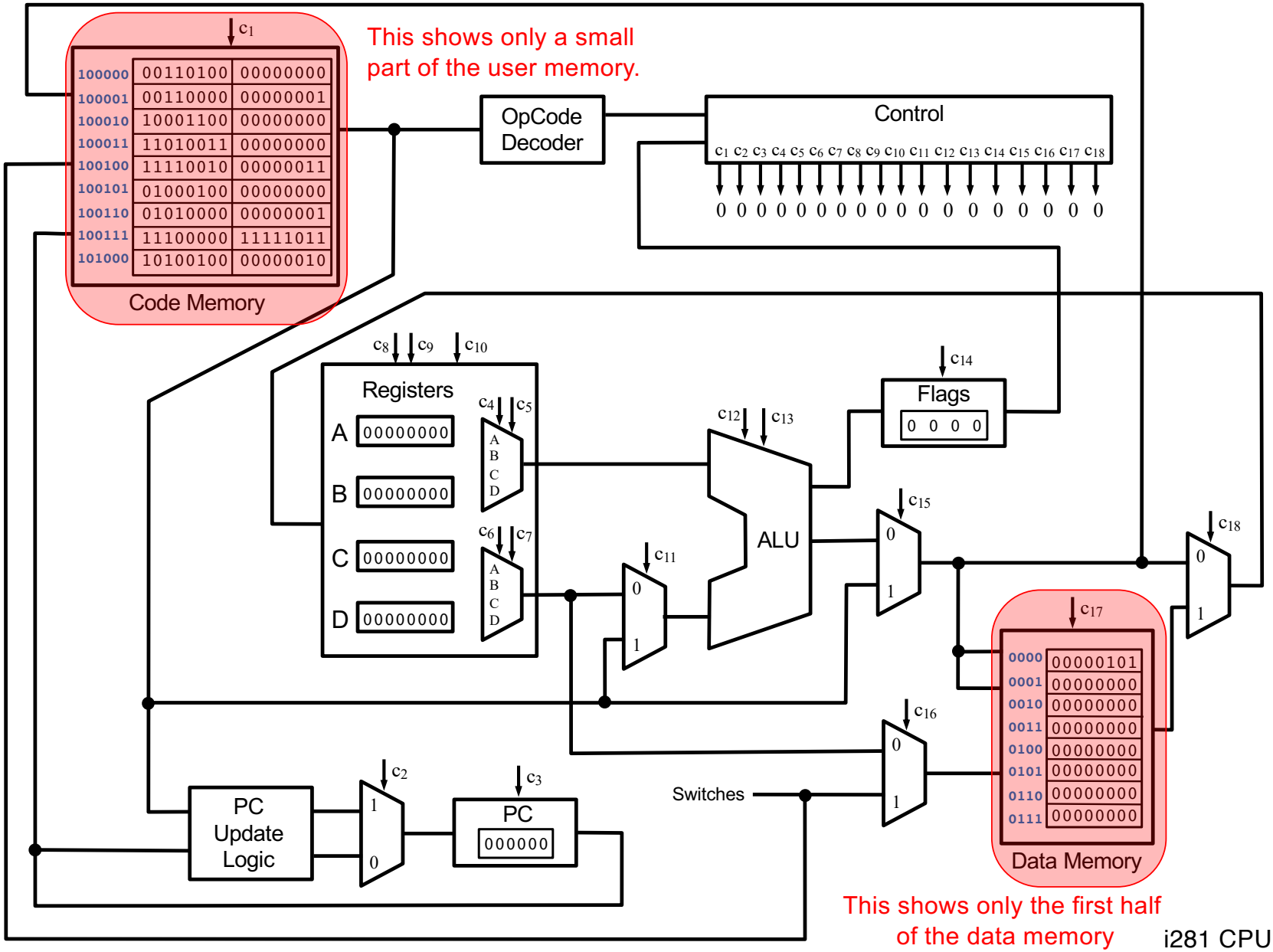
In this case the last 7 bits are used and each controls one of the 7 LEDs.

Code Memory Layout

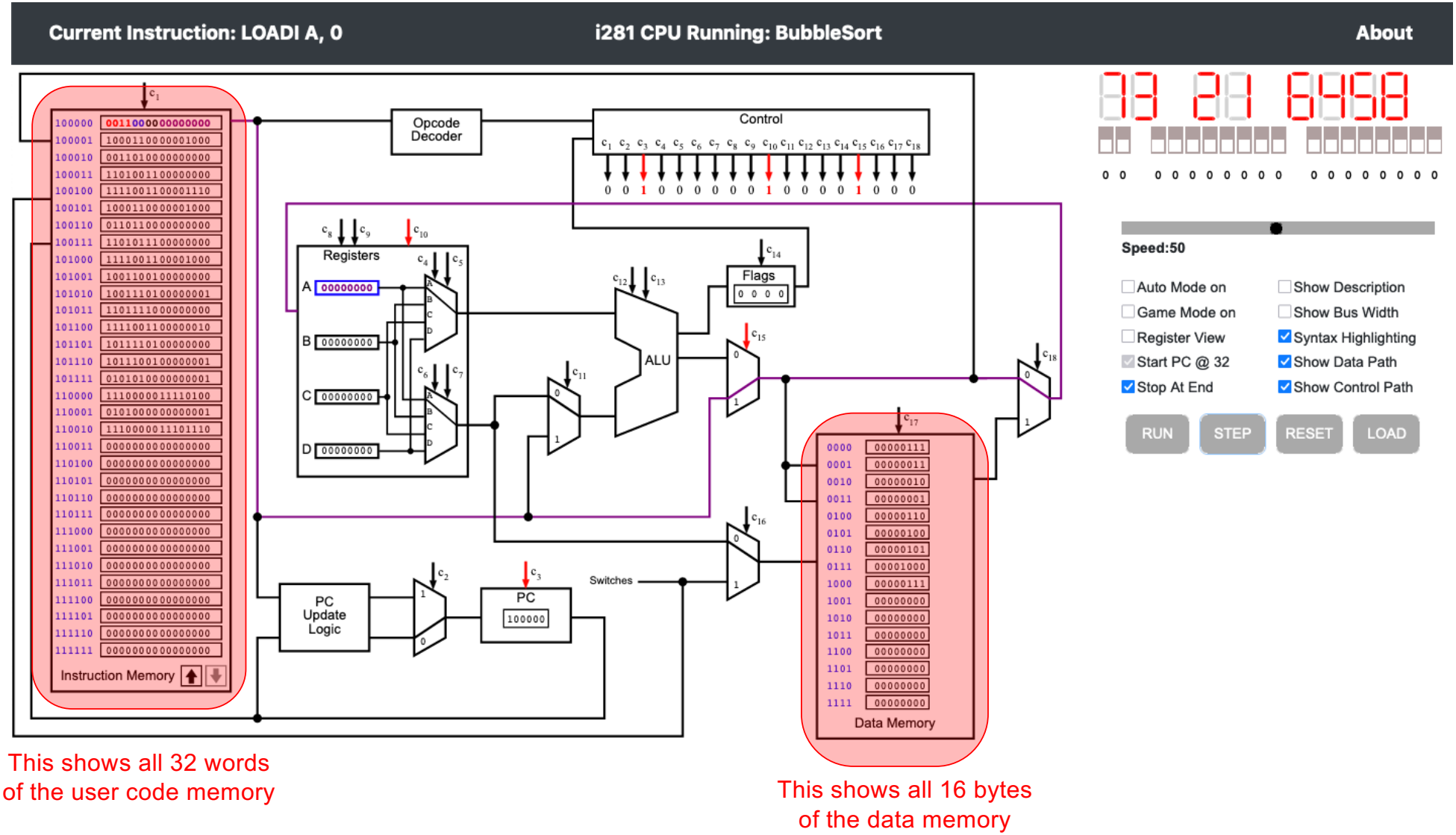
- Split into two parts

- BIOS Code Memory (addresses 0 to 31)
  - Read only memory
  - 32 x 16 bits

- User Code Memory (addresses 32 to 63)
  - Read only memory (in User mode)
  - Read/Write memory (in BIOS mode)
  - 32 x 16 bits


c1
This shows only a small part of the user memory.
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
OpCode Decoder
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
c14
Flags
0 0 0 0
c12 c13
ALU
c15
c11
c18
c17
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
c16
Switches
c2
c3
PC Update Logic
PC
000000
This shows only the first half of the data memory
i281 CPU

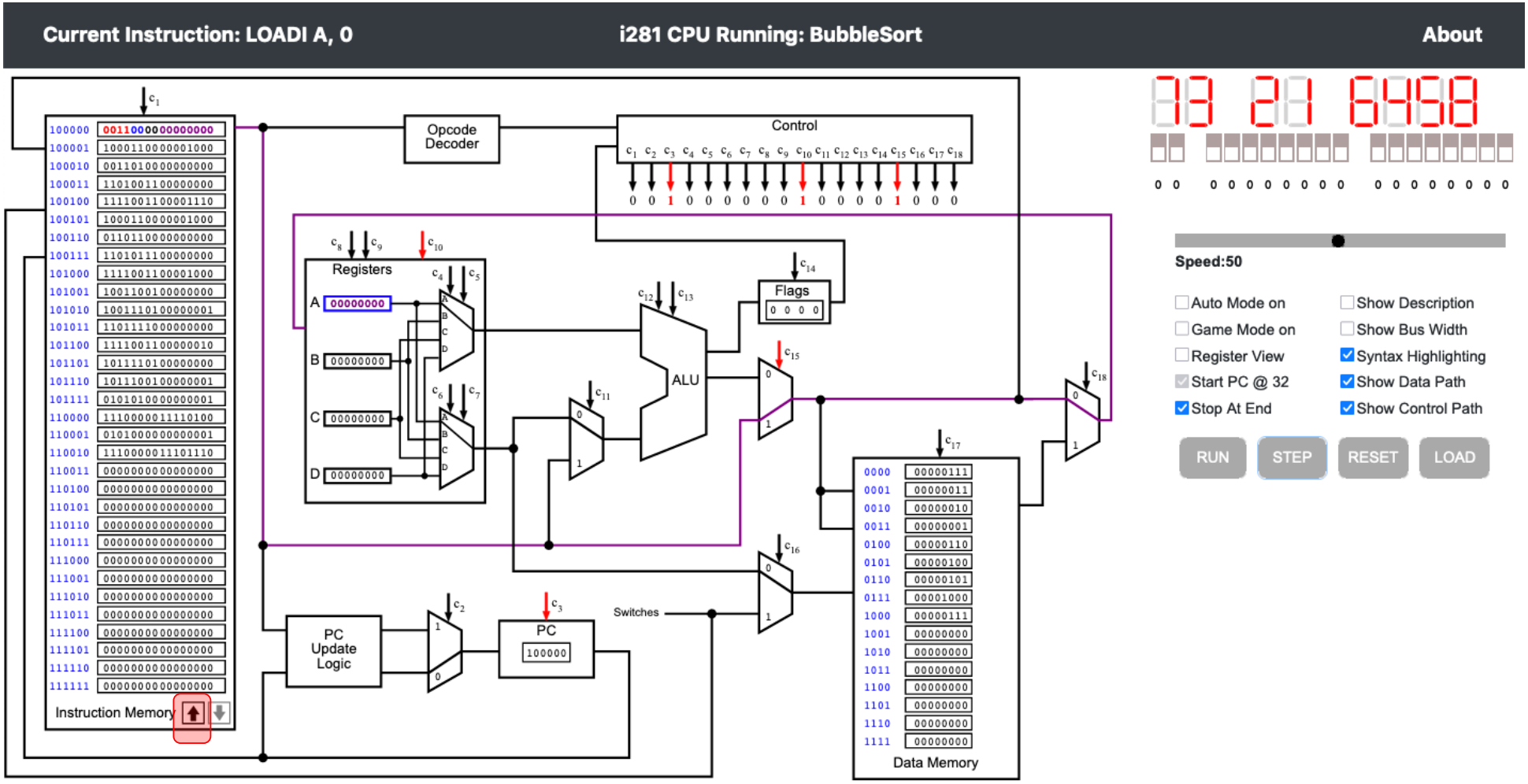
i281 Simulator



This shows all 32 words of the user code memory

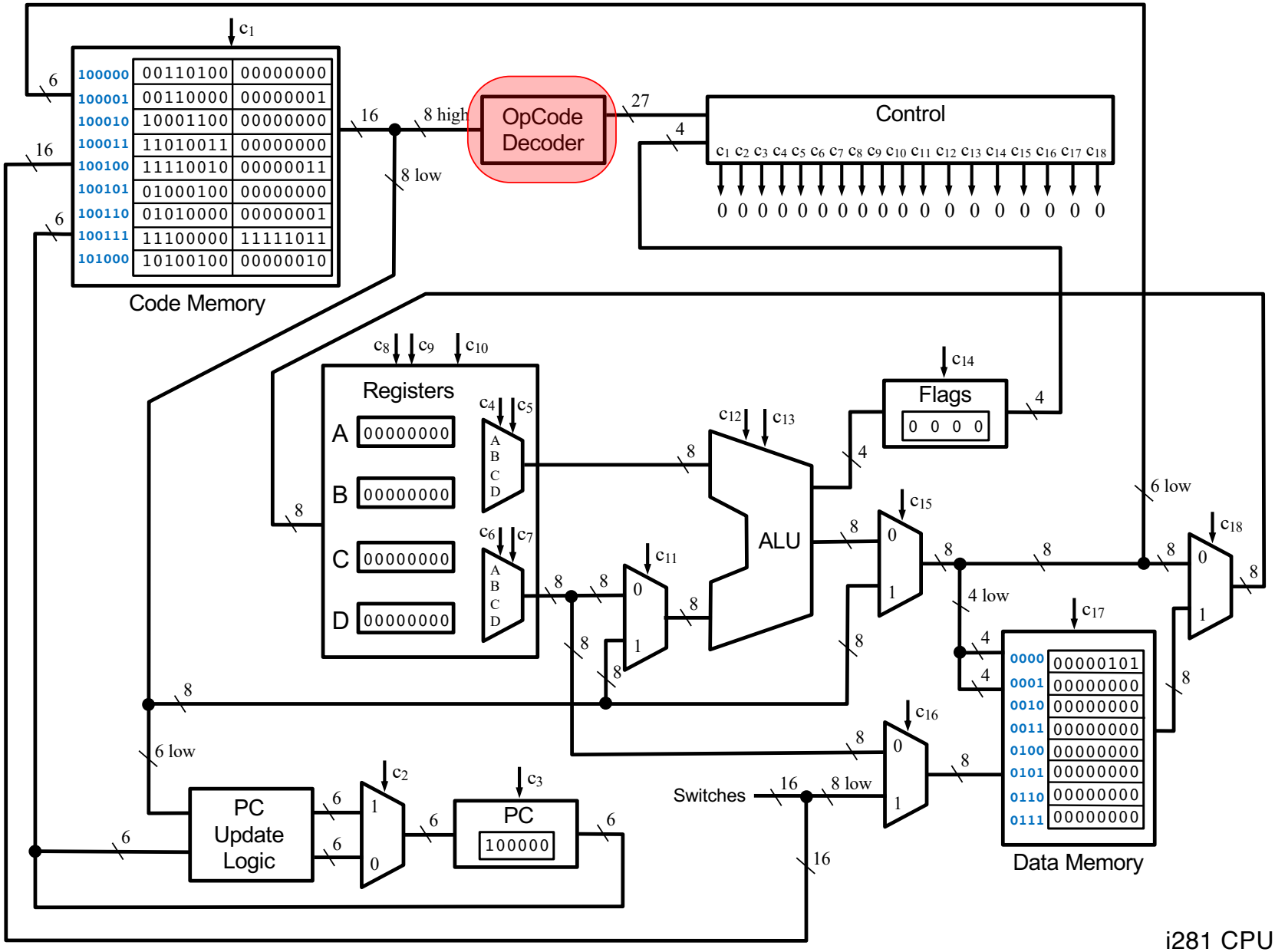
This shows all 16 bytes of the data memory

i281 Simulator



Use this button to see
the other 32 words
in the BIOS code memory

The OPCODE Decoder


c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
c12 c13
ALU
c14
Flags
0 0 0 0
c15
c11
c16
c17
c18
6 low
4 low
8 low
Switches
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
c2
c3
PC Update Logic
PC
100000
i281 CPU

The i281 Assembly Instructions

NOOP	NO OPeration
INPUTC	INPUT into Code memory
INPUTCF	INPUT into Code memory with oFfset
INPUTD	INPUT into Data memory
INPUTDF	INPUT into Data memory with oFfset
MOVE	MOVE the contents of one register into another
LOADI	LOAD Immediate value
LOADP	LOAD Pointer address
ADD	ADD two registers
ADDI	ADD an Immediate value to a register
SUB	SUBtract two registers
SUBI	SUBtract an Immediate value from a register
LOAD	LOAD from a data memory address into a register
LOADF	LOAD with an oFfset specified by another register
STORE	STORE a register into a data memory address
STOREF	STORE with an oFfset specified by another register
SHIFTL	SHIFT Left all bits in a register
SHIFTR	SHIFT Right all bits in a register
CMP	CoMPare the values in two registers
JUMP	JUMP unconditionally to a specified address
BRE	BRanch if Equal
BRZ	BRanch if Zero
BRNE	BRanch if Not Equal
BRNZ	BRanch if Not Zero
BRG	BRanch if Greater
BRGE	BRanch if Greater than or Equal

The i281 Assembly Instructions

NOOP	NO OPeration
INPUTC	INPUT into Code memory
INPUTCF	INPUT into Code memory with oFfset
INPUTD	INPUT into Data memory
INPUTDF	INPUT into Data memory with oFfset
MOVE	MOVE the contents of one register into another
LOADI	LOAD Immediate value
LOADP	LOAD Pointer address
ADD	ADD two registers
ADDI	ADD an Immediate value to a register
SUB	SUBtract two registers
SUBI	SUBtract an Immediate value from a register
LOAD	LOAD from a data memory address into a register
LOADF	LOAD with an oFfset specified by another register
STORE	STORE a register into a data memory address
STOREF	STORE with an oFfset specified by another register
SHIFTL	SHIFT Left all bits in a register
SHIFTR	SHIFT Right all bits in a register
CMP	CoMPare the values in two registers
JUMP	JUMP unconditionally to a specified address
BRE	BRanch if Equal
BRZ	BRanch if Zero
BRNE	BRanch if Not Equal
BRNZ	BRanch if Not Zero
BRG	BRanch if Greater
BRGE	BRanch if Greater than or Equal

There are only 26 OPCODEs

```
NOOP
INPUTC
INPUTCF
INPUTD
INPUTDF
MOVE
LOADI
LOADP
```

```
ADD
ADDI
SUB
SUBI
LOAD
LOADF
STORE
STOREF
```

```
SHIFTL
SHIFTR
CMP
JUMP
BRE
BRZ
BRNE
BRNZ
BRG
BRGE
```

There are only 26 OPCODEs

NOOP	ADD	SHIFTL
INPUTC	ADDI	SHIFTR
INPUTCF	SUB	CMP
INPUTD	SUBI	JUMP
INPUTDF	LOAD	BRE
MOVE	LOADF	BRZ
LOADI	STORE	BRNE
LOADP	STOREF	BRNZ
		BRG
		BRGE

All of these are available in the assembly language for this processor.
However, three pairs are aliased at the machine language level.

There are only ~~26~~ 25 OPCODEs

```
NOOP
INPUTC
INPUTCF
INPUTD
INPUTDF
MOVE
LOADI
LOADP
```

these two are aliased

```
ADD
ADDI
SUB
SUBI
LOAD
LOADF
STORE
STOREF
```

```
SHIFTL
SHIFTR
CMP
JUMP
BRE
BRZ
BRNE
BRNZ
BRG
BRGE
```

They have a different meaning in the assembly language, but the assembler maps them to the same machine language OPCODE.

There are only ~~26~~ 25 OPCODEs

NOOP
INPUTC
INPUTCF
INPUTD
INPUTDF
MOVE
LOADI/LOADP

ADD
ADDI
SUB
SUBI
LOAD
LOADF
STORE
STOREF

SHIFTL
SHIFTR
CMP
JUMP
BRE
BRZ
BRNE
BRNZ
BRG
BRGE

There are only ~~26~~ 24 OPCODEs

```
NOOP
INPUTC
INPUTCF
INPUTD
INPUTDF
MOVE
LOADI/LOADP
```

```
ADD
ADDI
SUB
SUBI
LOAD
LOADF
STORE
STOREF
```

```
SHIFTL
SHIFTR
CMP
JUMP
BRE
BRZ
BRNE
BRNZ
BRG
BRGE
```

these two are aliased

There are only ~~26~~ 23 OPCODEs

```
NOOP
INPUTC
INPUTCF
INPUTD
INPUTDF
MOVE
LOADI/LOADP
```

```
ADD
ADDI
SUB
SUBI
LOAD
LOADF
STORE
STOREF
```

```
SHIFTL
SHIFTR
CMP
JUMP
BRE
BRZ
BRNE
BRNZ
BRG
BRGE
```

these two are aliased

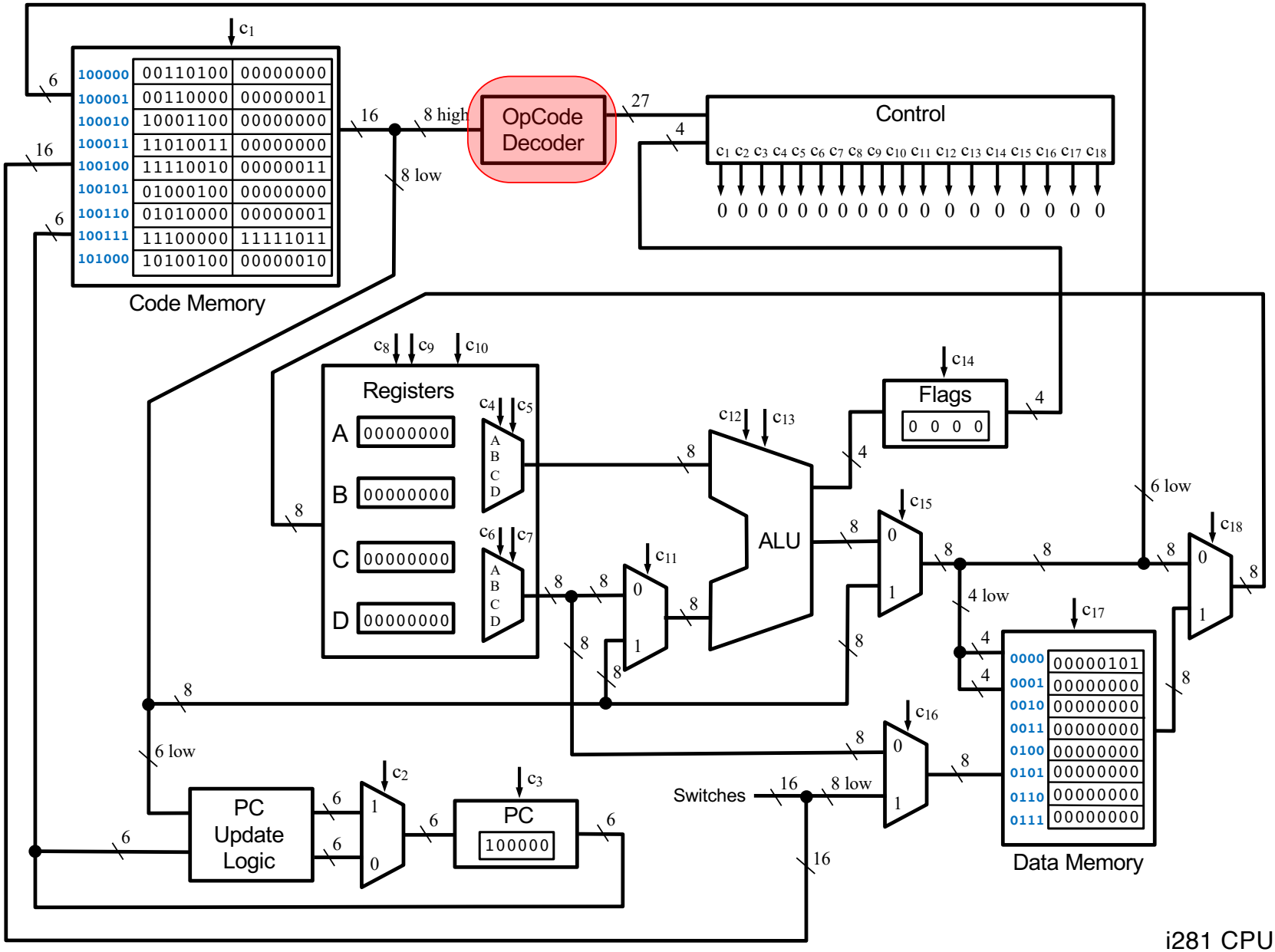
these two are aliased

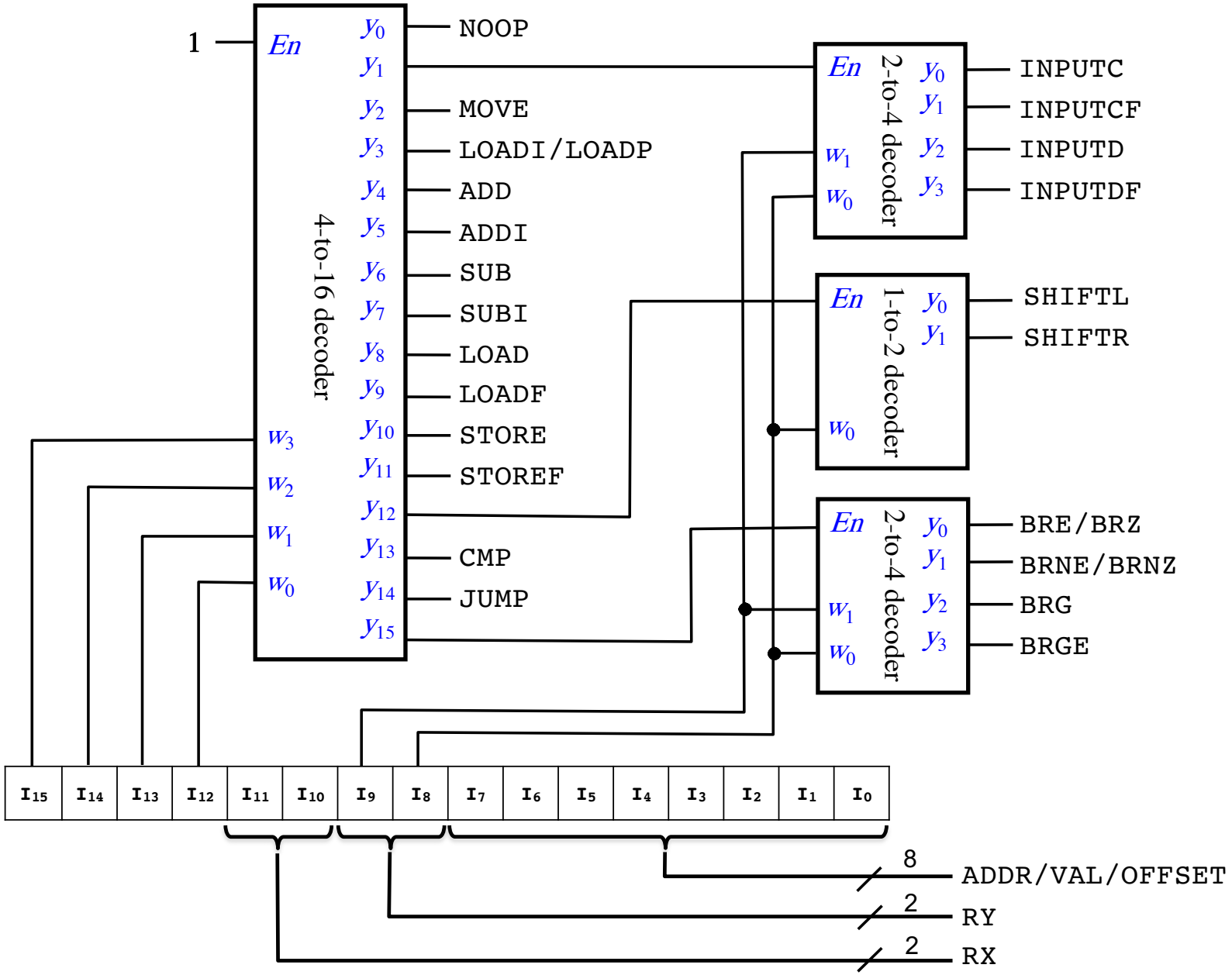
There are only 23 OPCODEs

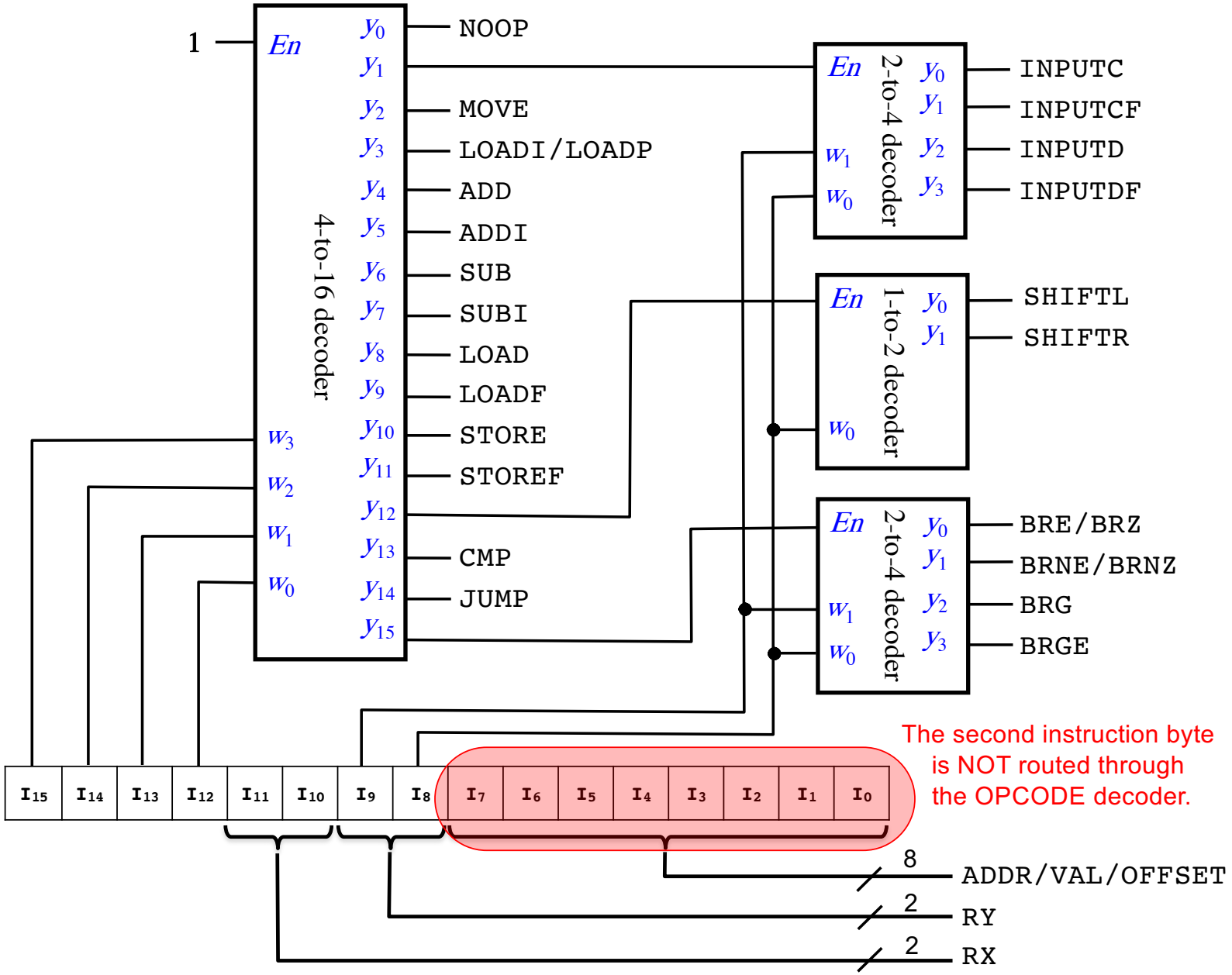
```
NOOP
INPUTC
INPUTCF
INPUTD
INPUTDF
MOVE
LOADI/LOADP
```

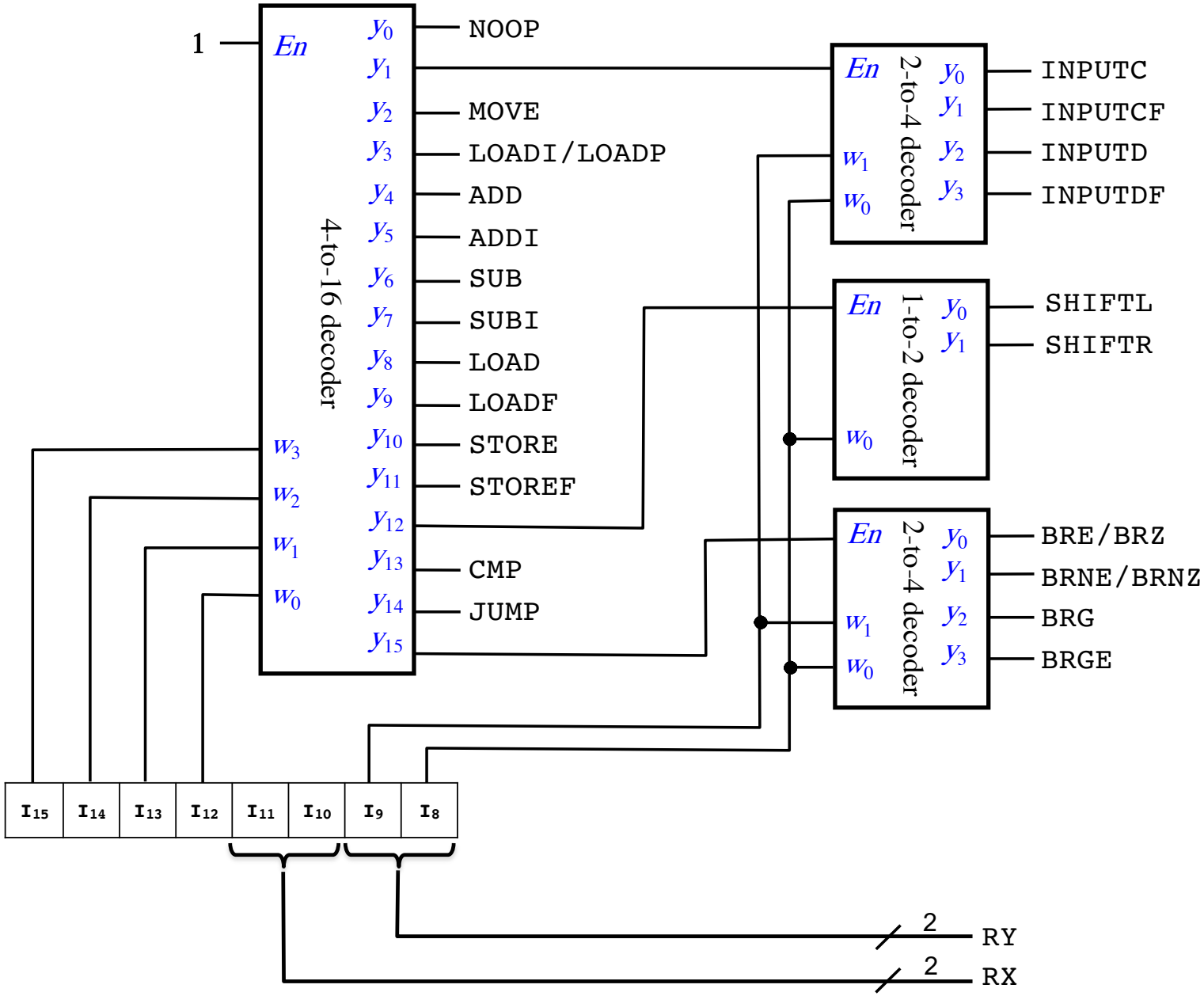
```
ADD
ADDI
SUB
SUBI
LOAD
LOADF
STORE
STOREF
```

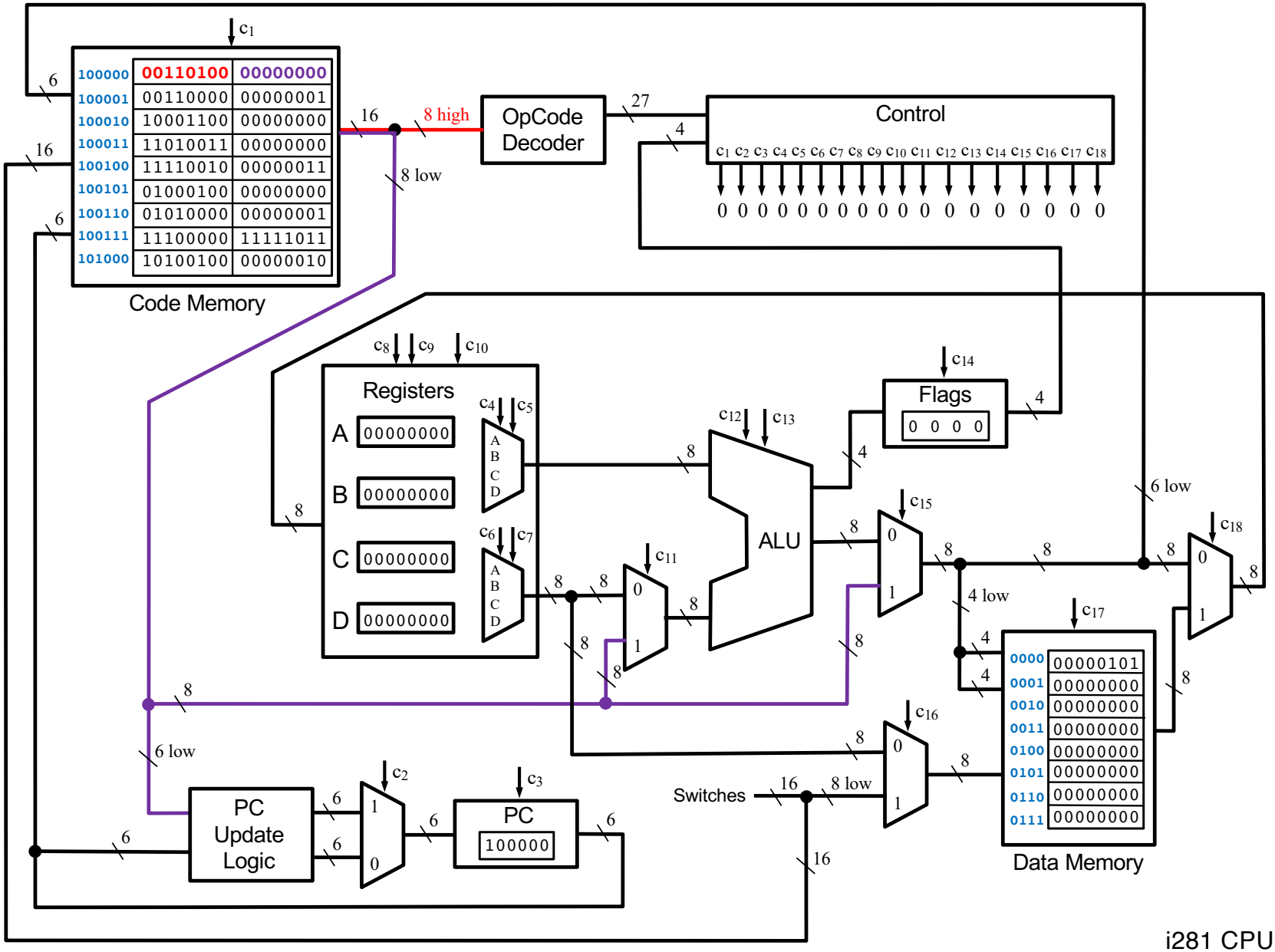
```
SHIFTL
SHIFTR
CMP
JUMP
BRE/BRZ
BRNE/BRNZ
BRG
BRGE
```

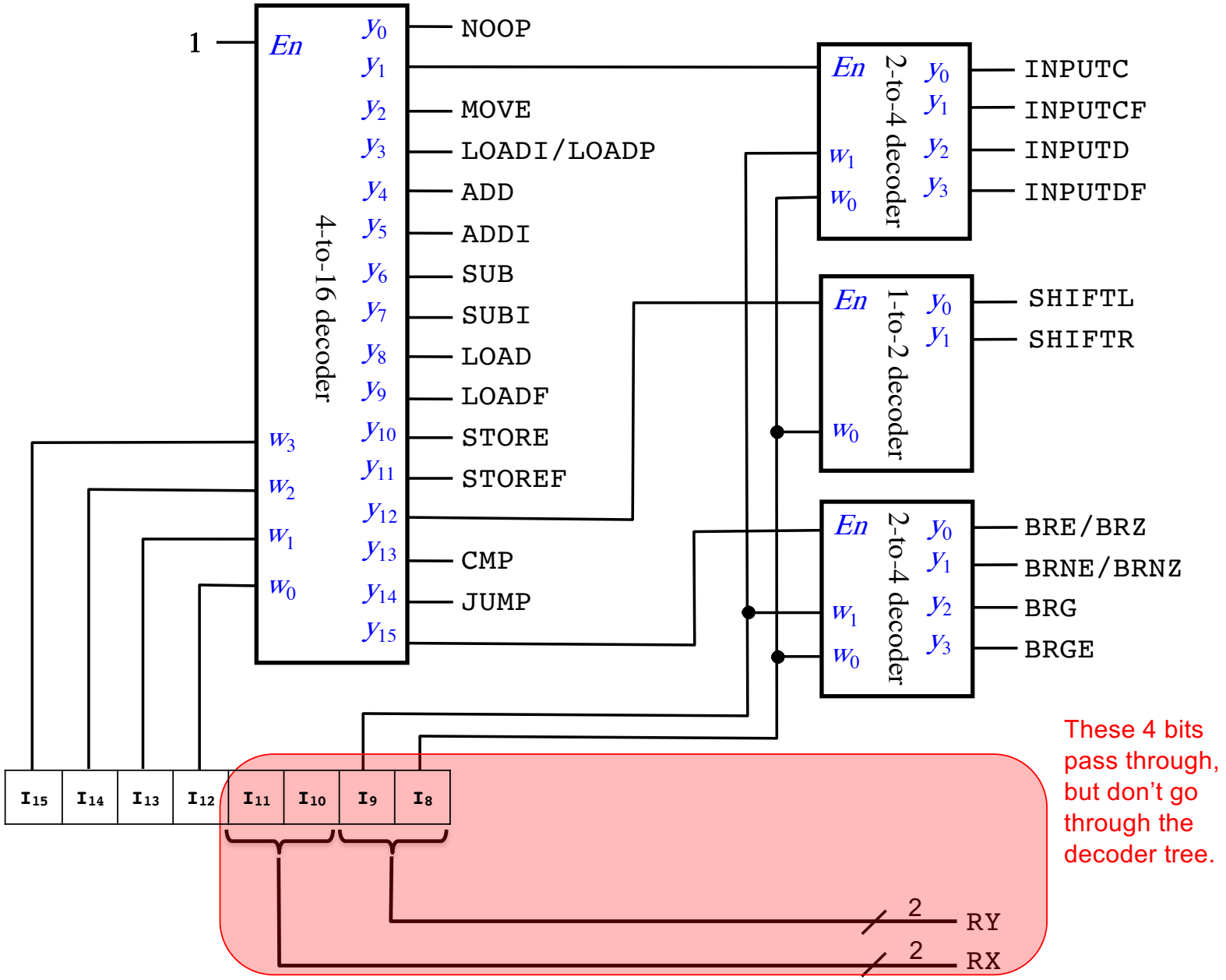

c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
c12 c13
ALU
c14
Flags
0 0 0 0
c15
c11
c16
c17
c18
6 low
4 low
8 low
Switches
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
c2
c3
PC Update Logic
PC
100000
i281 CPU

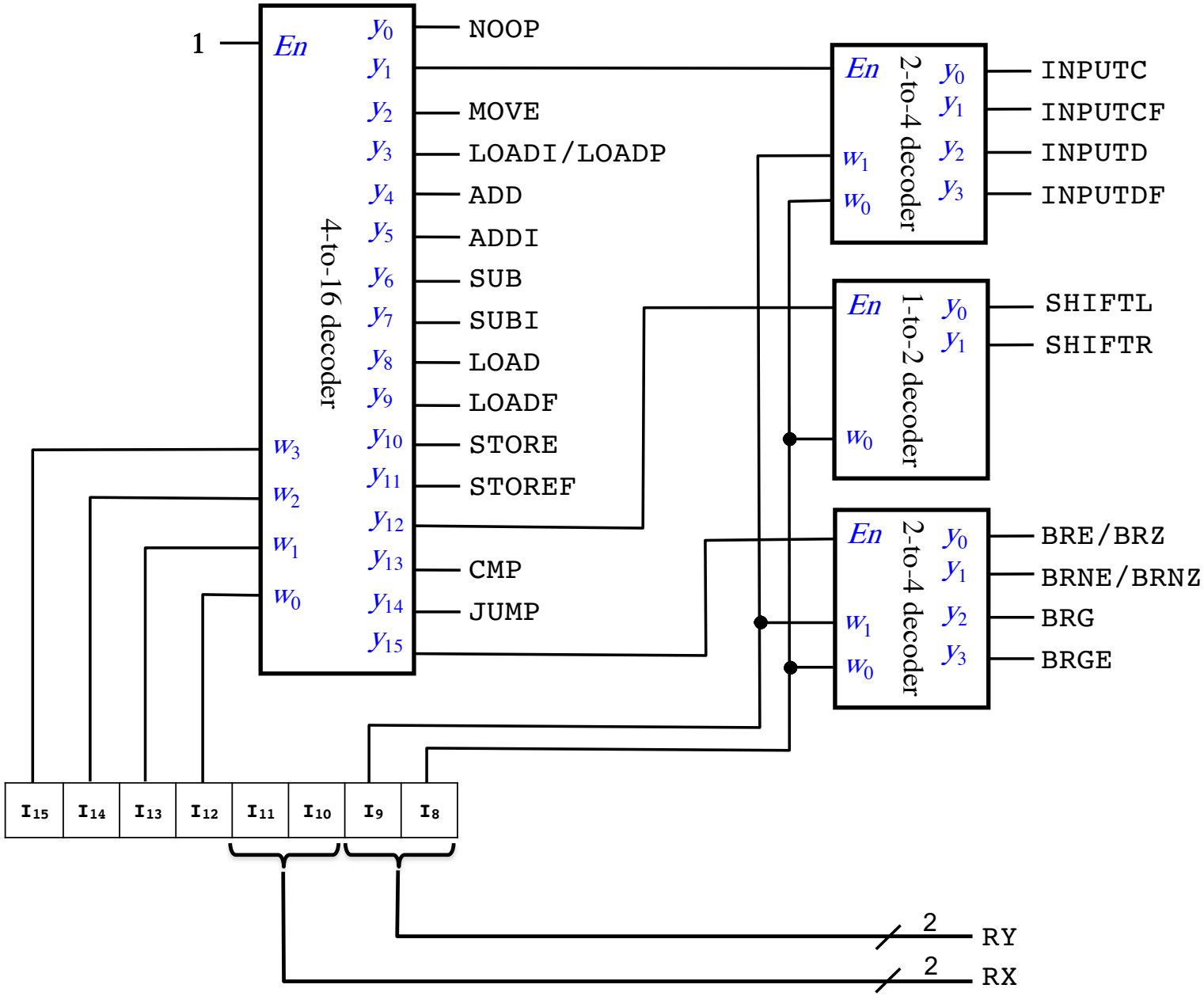

1
En
4-to-16 decoder
w_3
w_2
w_1
w_0
y_0 NOOP
y_1
y_2 MOVE
y_3 LOADI/LOADP
y_4 ADD
y_5 ADDI
y_6 SUB
y_7 SUBI
y_8 LOAD
y_9 LOADF
y_{10} STORE
y_{11} STOREF
y_{12}
y_{13} CMP
y_{14} JUMP
y_{15}
En
2-to-4 decoder
w_1
w_0
y_0 INPUTC
y_1 INPUTCF
y_2 INPUTD
y_3 INPUTDF
En
1-to-2 decoder
w_0
y_0 SHIFTL
y_1 SHIFTR
En
2-to-4 decoder
w_1
w_0
y_0 BRE/BRZ
y_1 BRNE/BRNZ
y_2 BRG
y_3 BRGE
I15 I14 I13 I12 I11 I10 I9 I8 I7 I6 I5 I4 I3 I2 I1 I0
8 ADDR/VAL/OFFSET
2 RY
2 RX

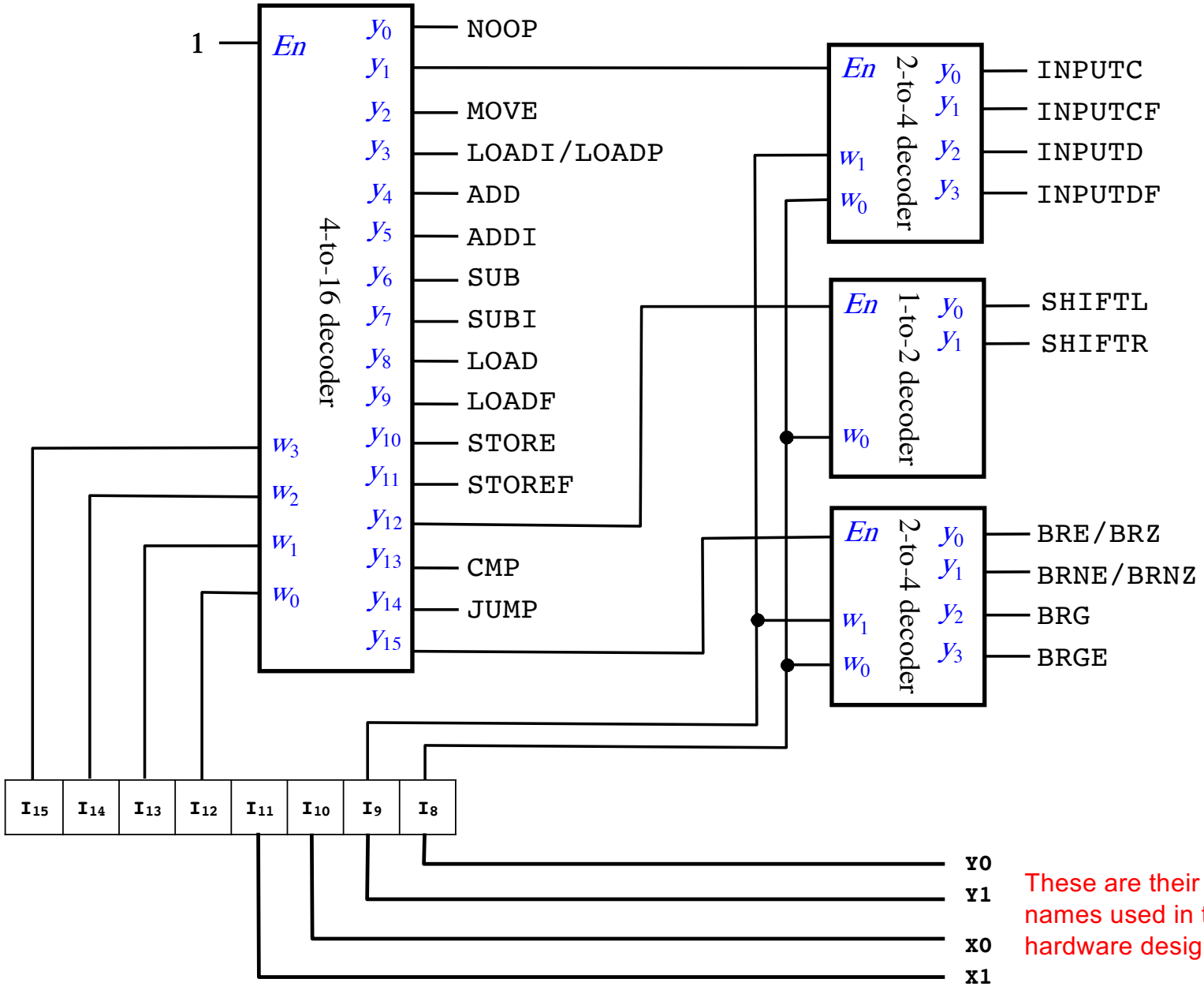

1
En
4-to-16 decoder
w_3
w_2
w_1
w_0
y_0 NOOP
y_1
y_2 MOVE
y_3 LOADI/LOADP
y_4 ADD
y_5 ADDI
y_6 SUB
y_7 SUBI
y_8 LOAD
y_9 LOADF
y_{10} STORE
y_{11} STOREF
y_{12}
y_{13} CMP
y_{14} JUMP
y_{15}
En
2-to-4 decoder
w_1
w_0
y_0 INPUTC
y_1 INPUTCF
y_2 INPUTD
y_3 INPUTDF
En
1-to-2 decoder
w_0
y_0 SHIFTL
y_1 SHIFTR
En
2-to-4 decoder
w_1
w_0
y_0 BRE/BRZ
y_1 BRNE/BRNZ
y_2 BRG
y_3 BRGE
I15 I14 I13 I12 I11 I10 I9 I8 I7 I6 I5 I4 I3 I2 I1 I0
The second instruction byte is NOT routed through the OPCODE decoder.
8 ADDR/VAL/OFFSET
2 RY
2 RX

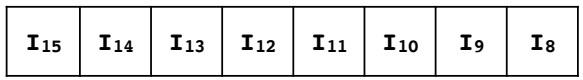

1
En
4-to-16 decoder
w_3
w_2
w_1
w_0
y_0 NOOP
y_1
y_2 MOVE
y_3 LOADI/LOADP
y_4 ADD
y_5 ADDI
y_6 SUB
y_7 SUBI
y_8 LOAD
y_9 LOADF
y_{10} STORE
y_{11} STOREF
y_{12}
y_{13} CMP
y_{14} JUMP
y_{15}
En
2-to-4 decoder
w_1
w_0
y_0 INPUTC
y_1 INPUTCF
y_2 INPUTD
y_3 INPUTDF
En
1-to-2 decoder
w_0
y_0 SHIFTL
y_1 SHIFTR
En
2-to-4 decoder
w_1
w_0
y_0 BRE/BRZ
y_1 BRNE/BRNZ
y_2 BRG
y_3 BRGE
I15 I14 I13 I12 I11 I10 I9 I8
2 RY
2 RX

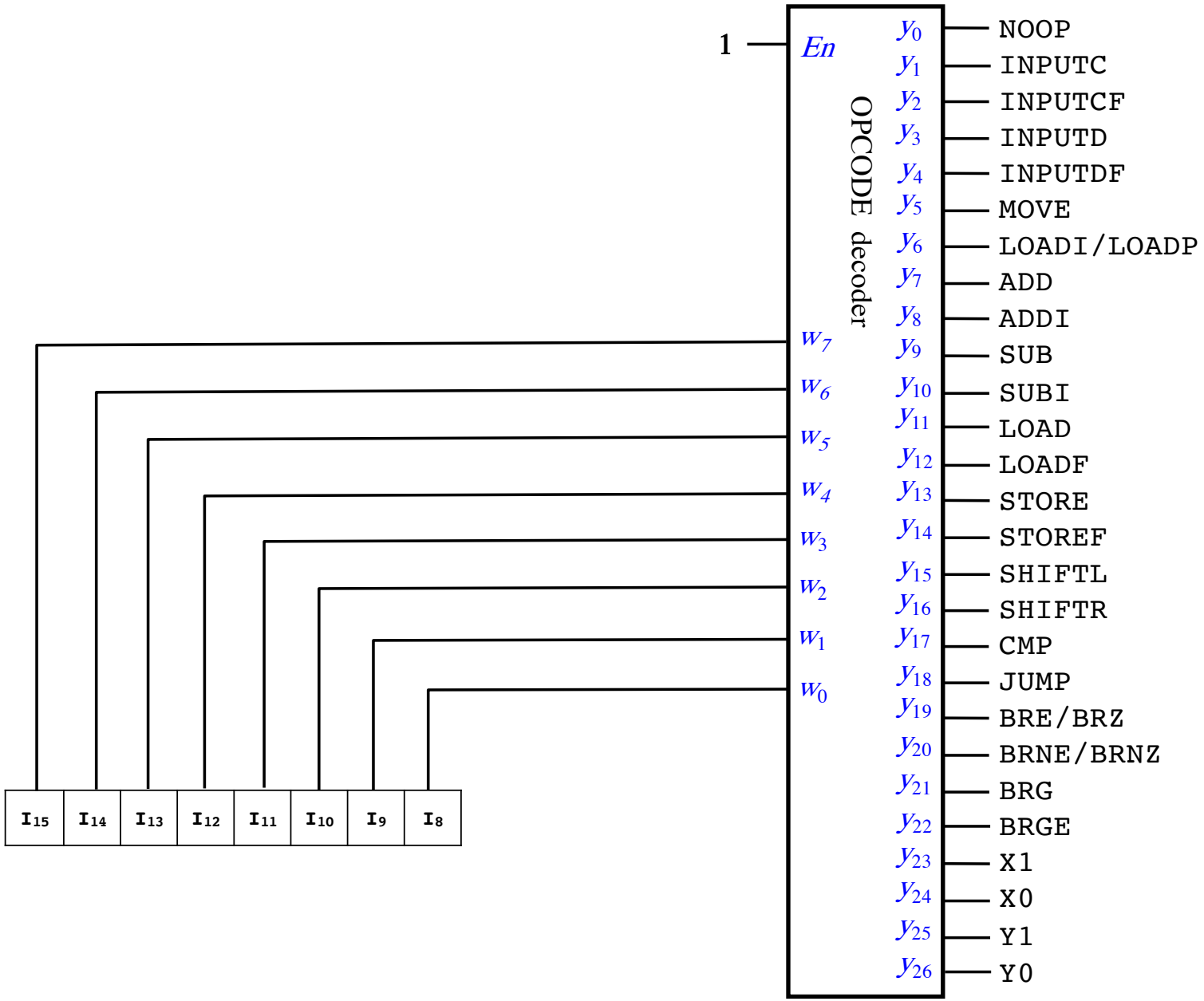

c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c12 c13
ALU
c14
Flags
0 0 0 0
4
c15
c11
c16
c17
c18
6 low
4 low
8 low
Switches
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
c2
c3
PC Update Logic
PC
100000
i281 CPU

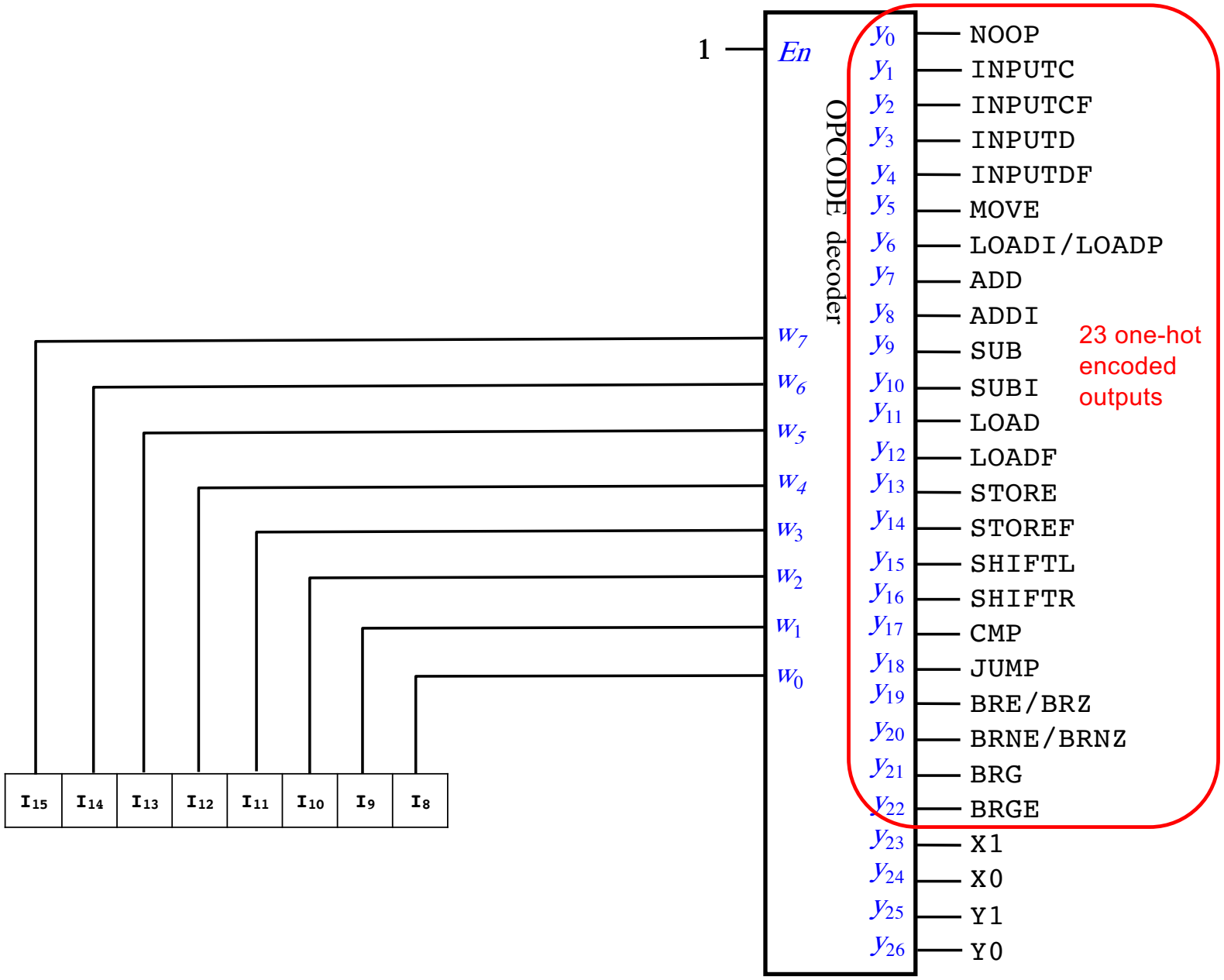

1
En
4-to-16 decoder
w_3
w_2
w_1
w_0
y_0 NOOP
y_1
y_2 MOVE
y_3 LOADI/LOADP
y_4 ADD
y_5 ADDI
y_6 SUB
y_7 SUBI
y_8 LOAD
y_9 LOADF
y_{10} STORE
y_{11} STOREF
y_{12}
y_{13} CMP
y_{14} JUMP
y_{15}
En
2-to-4 decoder
w_1
w_0
y_0 INPUTC
y_1 INPUTCF
y_2 INPUTD
y_3 INPUTDF
En
1-to-2 decoder
w_0
y_0 SHIFTL
y_1 SHIFTR
En
2-to-4 decoder
w_1
w_0
y_0 BRE/BRZ
y_1 BRNE/BRNZ
y_2 BRG
y_3 BRGE
I15 I14 I13 I12 I11 I10 I9 I8
2 RY
2 RX
These 4 bits pass through, but don't go through the decoder tree.

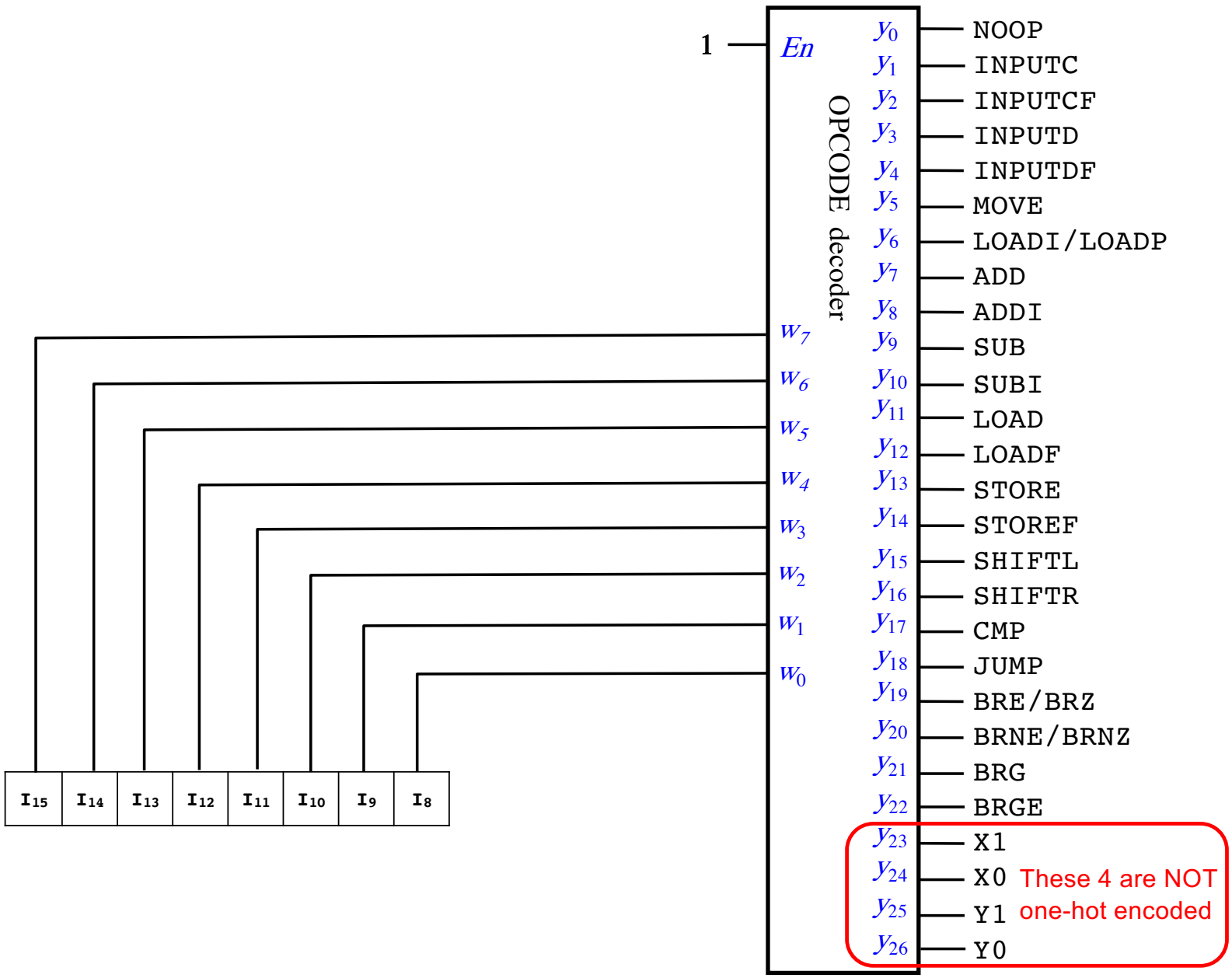

1
En
4-to-16 decoder
w_3
w_2
w_1
w_0
y_0 NOOP
y_1
y_2 MOVE
y_3 LOADI/LOADP
y_4 ADD
y_5 ADDI
y_6 SUB
y_7 SUBI
y_8 LOAD
y_9 LOADF
y_{10} STORE
y_{11} STOREF
y_{12}
y_{13} CMP
y_{14} JUMP
y_{15}
En
2-to-4 decoder
w_1
w_0
y_0 INPUTC
y_1 INPUTCF
y_2 INPUTD
y_3 INPUTDF
En
1-to-2 decoder
w_0
y_0 SHIFTL
y_1 SHIFTR
En
2-to-4 decoder
w_1
w_0
y_0 BRE/BRZ
y_1 BRNE/BRNZ
y_2 BRG
y_3 BRGE
I15 I14 I13 I12 I11 I10 I9 I8
2 RY
2 RX

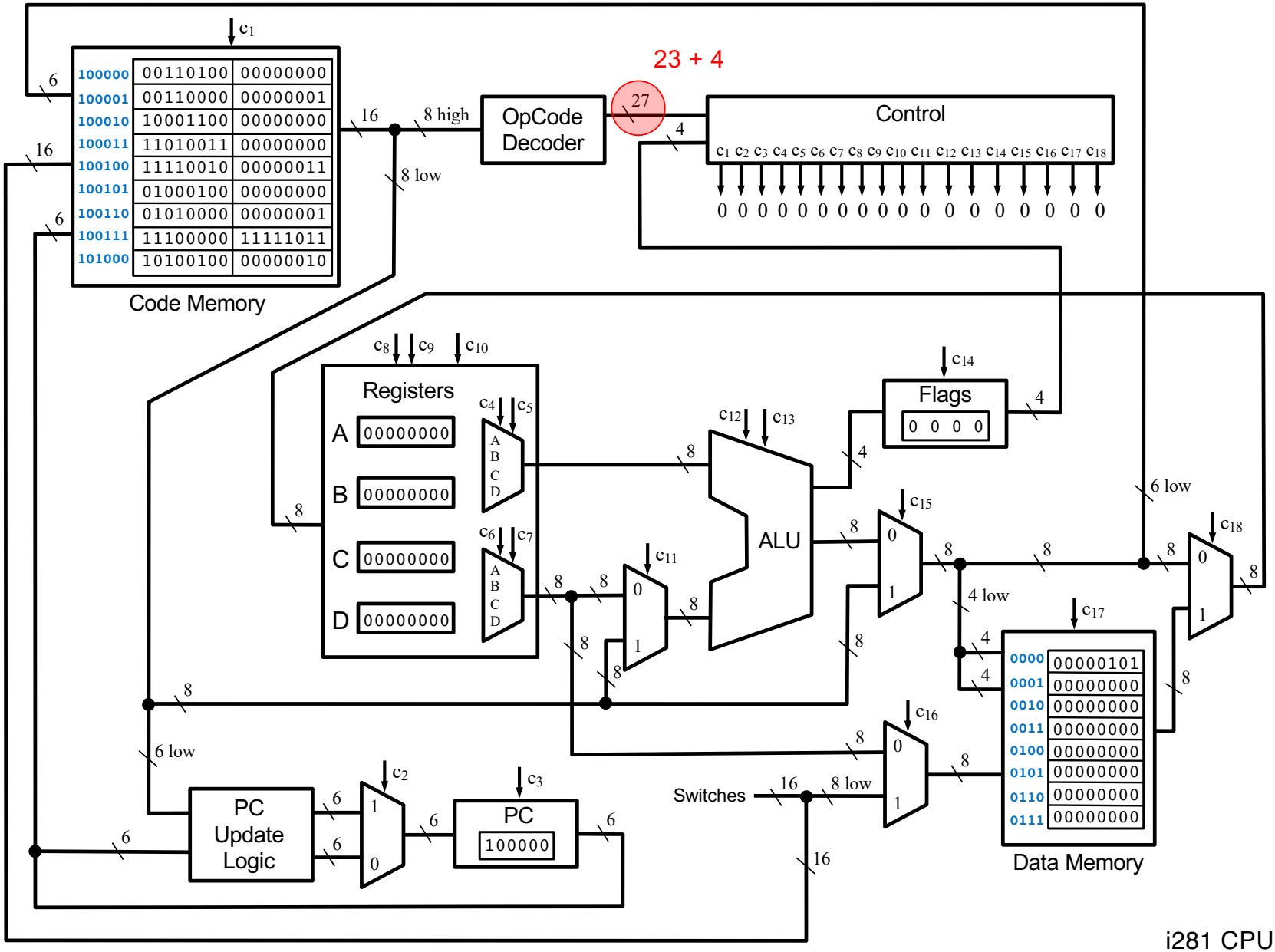

1
En
4-to-16 decoder
w_3
w_2
w_1
w_0
y_0 NOOP
y_1
y_2 MOVE
y_3 LOADI/LOADP
y_4 ADD
y_5 ADDI
y_6 SUB
y_7 SUBI
y_8 LOAD
y_9 LOADF
y_{10} STORE
y_{11} STOREF
y_{12}
y_{13} CMP
y_{14} JUMP
y_{15}
En
2-to-4 decoder
w_1
w_0
y_0 INPUTC
y_1 INPUTCF
y_2 INPUTD
y_3 INPUTDF
En
1-to-2 decoder
w_0
y_0 SHIFTL
y_1 SHIFTR
En
2-to-4 decoder
w_1
w_0
y_0 BRE/BRZ
y_1 BRNE/BRNZ
y_2 BRG
y_3 BRGE
I15 I14 I13 I12 I11 I10 I9 I8
Y0
Y1
X0
X1
These are their names used in the hardware design.


I15
I14
I13
I12
I11
I10
I9
I8

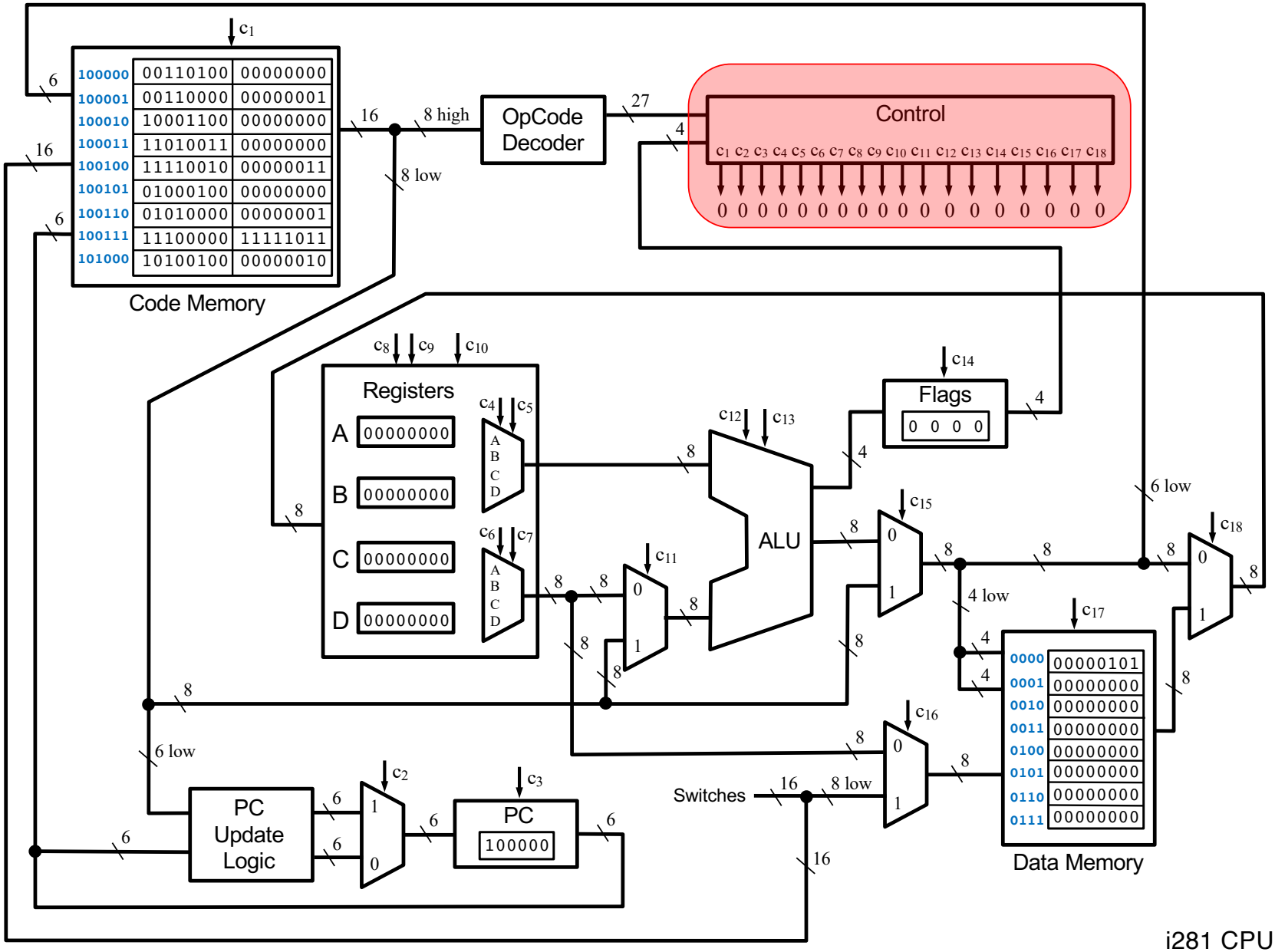

1
En
OPCODE decoder
w_7
w_6
w_5
w_4
w_3
w_2
w_1
w_0
y_0 NOOP
y_1 INPUTC
y_2 INPUTCF
y_3 INPUTD
y_4 INPUTDF
y_5 MOVE
y_6 LOADI/LOADP
y_7 ADD
y_8 ADDI
y_9 SUB
y_{10} SUBI
y_{11} LOAD
y_{12} LOADF
y_{13} STORE
y_{14} STOREF
y_{15} SHIFTL
y_{16} SHIFTR
y_{17} CMP
y_{18} JUMP
y_{19} BRE/BRZ
y_{20} BRNE/BRNZ
y_{21} BRG
y_{22} BRGE
y_{23} X1
y_{24} X0
y_{25} Y1
y_{26} Y0
I_{15} I_{14} I_{13} I_{12} I_{11} I_{10} I_9 I_8


1
En
OPCODE decoder
w_7
w_6
w_5
w_4
w_3
w_2
w_1
w_0
I_{15} I_{14} I_{13} I_{12} I_{11} I_{10} I_9 I_8
y_0 NOOP
y_1 INPUTC
y_2 INPUTCF
y_3 INPUTD
y_4 INPUTDF
y_5 MOVE
y_6 LOADI/LOADP
y_7 ADD
y_8 ADDI
y_9 SUB
y_{10} SUBI
y_{11} LOAD
y_{12} LOADF
y_{13} STORE
y_{14} STOREF
y_{15} SHIFTL
y_{16} SHIFTR
y_{17} CMP
y_{18} JUMP
y_{19} BRE/BRZ
y_{20} BRNE/BRNZ
y_{21} BRG
y_{22} BRGE
y_{23} X1
y_{24} X0
y_{25} Y1
y_{26} Y0
23 one-hot encoded outputs


1
En
OPCODE decoder
w_7
w_6
w_5
w_4
w_3
w_2
w_1
w_0
y_0 NOOP
y_1 INPUTC
y_2 INPUTCF
y_3 INPUTD
y_4 INPUTDF
y_5 MOVE
y_6 LOADI/LOADP
y_7 ADD
y_8 ADDI
y_9 SUB
y_{10} SUBI
y_{11} LOAD
y_{12} LOADF
y_{13} STORE
y_{14} STOREF
y_{15} SHIFTL
y_{16} SHIFTR
y_{17} CMP
y_{18} JUMP
y_{19} BRE/BRZ
y_{20} BRNE/BRNZ
y_{21} BRG
y_{22} BRGE
y_{23} X1
y_{24} X0
y_{25} Y1
y_{26} Y0
These 4 are NOT one-hot encoded
I_{15} I_{14} I_{13} I_{12} I_{11} I_{10} I_9 I_8


23 + 4
c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c12 c13
ALU
c14
Flags
0 0 0 0
4
c15
c11
c16
c17
c18
6 low
4 low
8 low
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
Switches
16
c2
c3
PC Update Logic
PC
100000
i281 CPU

The Control Logic


c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c14
Flags
0 0 0 0
4
c12 c13
ALU
c15
c11
c16
c17
c18
6 low
4 low
8 low
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
Switches
16
c2
c3
PC Update Logic
PC
100000
i281 CPU

	IMEM_WRITE_ENABLE	PROGRAM_COUNTER_MUX	PROGRAM_COUNTER_WRITE_ENABLE	REGISTERS_PORT0_SELECT1	REGISTERS_PORT0_SELECT0	REGISTERS_PORT1_SELECT1	REGISTERS_PORT1_SELECT0	REGISTERS_WRITE_SELECT1	REGISTERS_WRITE_SELECT0	REGISTERS_WRITE_ENABLE	ALU_SOURCE_MUX	ALU_SELECT1	ALU_SELECT0	FLAGS_WRITE_ENABLE	ALU_RESUT_MUX	DMEM_INPUT_MUX	DMEM_WRITE_ENABLE	REG_WRITEBACK_MUX
NOOP			1															
INPUTC	1		1												1			
INPUTCF	1		1	X1	X0						1	1						
INPUTD			1												1	1	1	
INPUTDF			1	X1	X0						1	1				1	1	
MOVE			1	Y1	Y0			X1	X0	1	1	1						
LOADI/LOADP			1					X1	X0	1					1			
ADD			1	X1	X0	Y1	Y0	X1	X0	1		1		1				
ADDI			1	X1	X0			X1	X0	1	1	1		1				
SUB			1	X1	X0	Y1	Y0	X1	X0	1		1	1	1				
SUBI			1	X1	X0			X1	X0	1	1	1	1	1				
LOAD			1					X1	X0	1					1			1
LOADF			1	Y1	Y0			X1	X0	1	1	1						1
STORE			1			X1	X0								1		1	
STOREF			1	Y1	Y0	X1	X0				1	1					1	
SHIFTL			1	X1	X0			X1	X0	1				1				
SHIFTR			1	X1	X0			X1	X0	1			1	1				
CMP			1	X1	X0	Y1	Y0					1	1	1				
JUMP		1	1															
BRE/BRZ		B1	1															
BRNE/BRNZ		B2	1															
BRG		B3	1															
BRGE		B4	1															

23 one-hot encoded lines from OPCODE decoder

	IMEM_WRITE_ENABLE	PROGRAM_COUNTER_MUX	PROGRAM_COUNTER_WRITE_ENABLE	REGISTERS_PORT0_SELECT1	REGISTERS_PORT0_SELECT0	REGISTERS_PORT1_SELECT1	REGISTERS_PORT1_SELECT0	REGISTERS_WRITE_SELECT1	REGISTERS_WRITE_SELECT0	REGISTERS_WRITE_ENABLE	ALU_SOURCE_MUX	ALU_SELECT1	ALU_SELECT0	FLAGS_WRITE_ENABLE	ALU_RESUT_MUX	DMEM_INPUT_MUX	DMEM_WRITE_ENABLE	REG_WRITEBACK_MUX
NOOP			1															
INPUTC	1		1												1			
INPUTCF	1		1	X1	X0						1	1						
INPUTD			1												1	1	1	
INPUTDF			1	X1	X0						1	1				1	1	
MOVE			1	Y1	Y0			X1	X0	1	1	1						
LOADI/LOADP			1					X1	X0	1					1			
ADD			1	X1	X0	Y1	Y0	X1	X0	1		1		1				
ADDI			1	X1	X0			X1	X0	1	1	1		1				
SUB			1	X1	X0	Y1	Y0	X1	X0	1		1	1	1				
SUBI			1	X1	X0			X1	X0	1	1	1	1	1				
LOAD			1					X1	X0	1					1			1
LOADF			1	Y1	Y0			X1	X0	1	1	1						1
STORE			1			X1	X0								1		1	
STOREF			1	Y1	Y0	X1	X0				1	1					1	
SHIFTL			1	X1	X0			X1	X0	1				1				
SHIFTR			1	X1	X0			X1	X0	1			1	1				
CMP			1	X1	X0	Y1	Y0					1	1	1				
JUMP		1	1															
BRE/BRZ		B1	1															
BRNE/BRNZ		B2	1															
BRG		B3	1															
BRGE		B4	1															

18 control lines

23 one-hot encoded OPCODEs

	c_1 IMEM_WRITE_ENABLE	c_2 PROGRAM_COUNTER_MUX	c_3 PROGRAM_COUNTER_WRITE_EN	c_4 REGISTERS_PORT0_SELECT1	c_5 REGISTERS_PORT0_SELECT0	c_6 REGISTERS_PORT1_SELECT1	c_7 REGISTERS_PORT1_SELECT0	c_8 REGISTERS_WRITE_SELECT1	c_9 REGISTERS_WRITE_SELECT0	c_{10} REGISTERS_WRITE_ENABLE	c_{11} ALU_SOURCE_MUX	c_{12} ALU_SELECT1	c_{13} ALU_SELECT0	c_{14} FLAGS_WRITE_ENABLE	c_{15} ALU_RESUT_MUX	c_{16} DMEM_INPUT_MUX	c_{17} DMEM_WRITE_ENABLE	c_{18} REG_WRITEBACK_MUX
NOOP			1															
INPUTC	1		1												1			
INPUTCF	1		1	X1	X0						1	1						
INPUTD			1												1	1	1	
INPUTDF			1	X1	X0						1	1				1	1	
MOVE			1	Y1	Y0			X1	X0	1	1	1						
LOADI/LOADP			1					X1	X0	1					1			
ADD			1	X1	X0	Y1	Y0	X1	X0	1		1		1				
ADDI			1	X1	X0			X1	X0	1	1	1		1				
SUB			1	X1	X0	Y1	Y0	X1	X0	1		1	1	1				
SUBI			1	X1	X0			X1	X0	1	1	1	1	1				
LOAD			1					X1	X0	1					1			1
LOADF			1	Y1	Y0			X1	X0	1	1	1						1
STORE			1			X1	X0								1		1	
STOREF			1	Y1	Y0	X1	X0				1	1					1	
SHIFTL			1	X1	X0			X1	X0	1				1				
SHIFTR			1	X1	X0			X1	X0	1			1	1				
CMP			1	X1	X0	Y1	Y0					1	1	1				
JUMP		1	1															
BRE/BRZ		B1	1															
BRNE/BRNZ		B2	1															
BRG		B3	1															
BRGE		B4	1															

	c_1 IMEM_WRITE_ENABLE	c_2 PROGRAM_COUNTER_MUX	c_3 PROGRAM_COUNTER_WRITE_EN	c_4 REGISTERS_PORT0_SELECT1	c_5 REGISTERS_PORT0_SELECT0	c_6 REGISTERS_PORT1_SELECT1	c_7 REGISTERS_PORT1_SELECT0	c_8 REGISTERS_WRITE_SELECT1	c_9 REGISTERS_WRITE_SELECT0	c_{10} REGISTERS_WRITE_ENABLE	c_{11} ALU_SOURCE_MUX	c_{12} ALU_SELECT1	c_{13} ALU_SELECT0	c_{14} FLAGS_WRITE_ENABLE	c_{15} ALU_RESUT_MUX	c_{16} DMEM_INPUT_MUX	c_{17} DMEM_WRITE_ENABLE	c_{18} REG_WRITEBACK_MUX
NOOP			1															
INPUTC	1		1												1			
INPUTCF	1		1	X1	X0						1	1						
INPUTD			1												1	1	1	
INPUTDF			1	X1	X0						1	1				1	1	
MOVE			1	Y1	Y0			X1	X0	1	1	1						
LOADI/LOADP			1					X1	X0	1					1			
ADD			1	X1	X0	Y1	Y0	X1	X0	1		1		1				
ADDI			1	X1	X0			X1	X0	1	1	1		1				
SUB			1	X1	X0	Y1	Y0	X1	X0	1		1	1	1				
SUBI			1	X1	X0			X1	X0	1	1	1	1	1				
LOAD			1					X1	X0	1					1			1
LOADF			1	Y1	Y0			X1	X0	1	1	1						1
STORE			1			X1	X0								1		1	
STOREF			1	Y1	Y0	X1	X0				1	1					1	
SHIFTL			1	X1	X0			X1	X0	1				1				
SHIFTR			1	X1	X0			X1	X0	1			1	1				
CMP			1	X1	X0	Y1	Y0					1	1	1				
JUMP		1	1															
BRE/BRZ		B1	1															
BRNE/BRNZ		B2	1															
BRG		B3	1															
BRGE		B4	1															

Taken from these bits of the instruction

I_{15}	I_{14}	I_{13}	I_{12}	I_{11}	I_{10}	I_9	I_8	I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0
				X_1	X_0										

	c_1 IMEM_WRITE_ENABLE	c_2 PROGRAM_COUNTER_MUX	c_3 PROGRAM_COUNTER_WRITE_EN	c_4 REGISTERS_PORT0_SELECT1	c_5 REGISTERS_PORT0_SELECT0	c_6 REGISTERS_PORT1_SELECT1	c_7 REGISTERS_PORT1_SELECT0	c_8 REGISTERS_WRITE_SELECT1	c_9 REGISTERS_WRITE_SELECT0	c_{10} REGISTERS_WRITE_ENABLE	c_{11} ALU_SOURCE_MUX	c_{12} ALU_SELECT1	c_{13} ALU_SELECT0	c_{14} FLAGS_WRITE_ENABLE	c_{15} ALU_RESUT_MUX	c_{16} DMEM_INPUT_MUX	c_{17} DMEM_WRITE_ENABLE	c_{18} REG_WRITEBACK_MUX
NOOP			1															
INPUTC	1		1												1			
INPUTCF	1		1	X1	X0						1	1						
INPUTD			1												1	1	1	
INPUTDF			1	X1	X0						1	1				1	1	
MOVE			1	Y1	Y0			X1	X0	1	1	1						
LOADI/LOADP			1					X1	X0	1					1			
ADD			1	X1	X0	Y1	Y0	X1	X0	1		1		1				
ADDI			1	X1	X0			X1	X0	1	1	1		1				
SUB			1	X1	X0	Y1	Y0	X1	X0	1		1	1	1				
SUBI			1	X1	X0			X1	X0	1	1	1	1	1				
LOAD			1					X1	X0	1					1			1
LOADF			1	Y1	Y0			X1	X0	1	1	1						1
STORE			1			X1	X0								1		1	
STOREF			1	Y1	Y0	X1	X0				1	1					1	
SHIFTL			1	X1	X0			X1	X0	1				1				
SHIFTR			1	X1	X0			X1	X0	1			1	1				
CMP			1	X1	X0	Y1	Y0					1	1	1				
JUMP		1	1															
BRE/BRZ		B1	1															
BRNE/BRNZ		B2	1															
BRG		B3	1															
BRGE		B4	1															

Taken from these bits of the instruction

I_{15}	I_{14}	I_{13}	I_{12}	I_{11}	I_{10}	I_9	I_8	I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0
				X_1	X_0										

	c_1 IMEM_WRITE_ENABLE	c_2 PROGRAM_COUNTER_MUX	c_3 PROGRAM_COUNTER_WRITE_EN	c_4 REGISTERS_PORT0_SELECT1	c_5 REGISTERS_PORT0_SELECT0	c_6 REGISTERS_PORT1_SELECT1	c_7 REGISTERS_PORT1_SELECT0	c_8 REGISTERS_WRITE_SELECT1	c_9 REGISTERS_WRITE_SELECT0	c_{10} REGISTERS_WRITE_ENABLE	c_{11} ALU_SOURCE_MUX	c_{12} ALU_SELECT1	c_{13} ALU_SELECT0	c_{14} FLAGS_WRITE_ENABLE	c_{15} ALU_RESUT_MUX	c_{16} DMEM_INPUT_MUX	c_{17} DMEM_WRITE_ENABLE	c_{18} REG_WRITEBACK_MUX
NOOP			1															
INPUTC	1		1												1			
INPUTCF	1		1	X1	X0						1	1						
INPUTD			1												1	1	1	
INPUTDF			1	X1	X0						1	1				1	1	
MOVE			1	Y1	Y0			X1	X0	1	1	1						
LOADI/LOADP			1					X1	X0	1					1			
ADD			1	X1	X0	Y1	Y0	X1	X0	1		1		1				
ADDI			1	X1	X0			X1	X0	1	1	1		1				
SUB			1	X1	X0	Y1	Y0	X1	X0	1		1	1	1				
SUBI			1	X1	X0			X1	X0	1	1	1	1	1				
LOAD			1					X1	X0	1					1			1
LOADF			1	Y1	Y0			X1	X0	1	1	1						1
STORE			1			X1	X0								1		1	
STOREF			1	Y1	Y0	X1	X0				1	1					1	
SHIFTL			1	X1	X0			X1	X0	1				1				
SHIFTR			1	X1	X0			X1	X0	1			1	1				
CMP			1	X1	X0	Y1	Y0					1	1	1				
JUMP		1	1															
BRE/BRZ		B1	1															
BRNE/BRNZ		B2	1															
BRG		B3	1															
BRGE		B4	1															

Taken from these bits of the instruction

I_{15}	I_{14}	I_{13}	I_{12}	I_{11}	I_{10}	I_9	I_8	I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0
						Y_1	Y_0								

	c_1 IMEM_WRITE_ENABLE	c_2 PROGRAM_COUNTER_MUX	c_3 PROGRAM_COUNTER_WRITE_EN	c_4 REGISTERS_PORT0_SELECT1	c_5 REGISTERS_PORT0_SELECT0	c_6 REGISTERS_PORT1_SELECT1	c_7 REGISTERS_PORT1_SELECT0	c_8 REGISTERS_WRITE_SELECT1	c_9 REGISTERS_WRITE_SELECT0	c_{10} REGISTERS_WRITE_ENABLE	c_{11} ALU_SOURCE_MUX	c_{12} ALU_SELECT1	c_{13} ALU_SELECT0	c_{14} FLAGS_WRITE_ENABLE	c_{15} ALU_RESUT_MUX	c_{16} DMEM_INPUT_MUX	c_{17} DMEM_WRITE_ENABLE	c_{18} REG_WRITEBACK_MUX
NOOP			1															
INPUTC	1		1												1			
INPUTCF	1		1	X1	X0						1	1						
INPUTD			1												1	1	1	
INPUTDF			1	X1	X0						1	1				1	1	
MOVE			1	Y1	Y0			X1	X0	1	1	1						
LOADI/LOADP			1					X1	X0	1					1			
ADD			1	X1	X0	Y1	Y0	X1	X0	1		1		1				
ADDI			1	X1	X0			X1	X0	1	1	1		1				
SUB			1	X1	X0	Y1	Y0	X1	X0	1		1	1	1				
SUBI			1	X1	X0			X1	X0	1	1	1	1	1				
LOAD			1					X1	X0	1					1			1
LOADF			1	Y1	Y0			X1	X0	1	1	1						1
STORE			1			X1	X0								1		1	
STOREF			1	Y1	Y0	X1	X0				1	1					1	
SHIFTL			1	X1	X0			X1	X0	1				1				
SHIFTR			1	X1	X0			X1	X0	1			1	1				
CMP			1	X1	X0	Y1	Y0					1	1	1				
JUMP		1	1															
BRE/BRZ		B1	1															
BRNE/BRNZ		B2	1															
BRG		B3	1															
BRGE		B4	1															

Taken from these bits of the instruction

I_{15}	I_{14}	I_{13}	I_{12}	I_{11}	I_{10}	I_9	I_8	I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0
						Y_1	Y_0								

	c_1 IMEM_WRITE_ENABLE	c_2 PROGRAM_COUNTER_MUX	c_3 PROGRAM_COUNTER_WRITE_EN	c_4 REGISTERS_PORT0_SELECT1	c_5 REGISTERS_PORT0_SELECT0	c_6 REGISTERS_PORT1_SELECT1	c_7 REGISTERS_PORT1_SELECT0	c_8 REGISTERS_WRITE_SELECT1	c_9 REGISTERS_WRITE_SELECT0	c_{10} REGISTERS_WRITE_ENABLE	c_{11} ALU_SOURCE_MUX	c_{12} ALU_SELECT1	c_{13} ALU_SELECT0	c_{14} FLAGS_WRITE_ENABLE	c_{15} ALU_RESUT_MUX	c_{16} DMEM_INPUT_MUX	c_{17} DMEM_WRITE_ENABLE	c_{18} REG_WRITEBACK_MUX
NOOP			1															
INPUTC	1		1												1			
INPUTCF	1		1	X1	X0						1	1						
INPUTD			1												1	1	1	
INPUTDF			1	X1	X0						1	1				1	1	
MOVE			1	Y1	Y0			X1	X0	1	1	1						
LOADI/LOADP			1					X1	X0	1					1			
ADD			1	X1	X0	Y1	Y0	X1	X0	1		1		1				
ADDI			1	X1	X0			X1	X0	1	1	1		1				
SUB			1	X1	X0	Y1	Y0	X1	X0	1		1	1	1				
SUBI			1	X1	X0			X1	X0	1	1	1	1	1				
LOAD			1					X1	X0	1					1			1
LOADF			1	Y1	Y0			X1	X0	1	1	1						1
STORE			1			X1	X0								1		1	
STOREF			1	Y1	Y0	X1	X0				1	1					1	
SHIFTL			1	X1	X0			X1	X0	1				1				
SHIFTR			1	X1	X0			X1	X0	1			1	1				
CMP			1	X1	X0	Y1	Y0					1	1	1				
JUMP		1	1															
BRE/BRZ		B1	1															
BRNE/BRNZ		B2	1															
BRG		B3	1															
BRGE		B4	1															

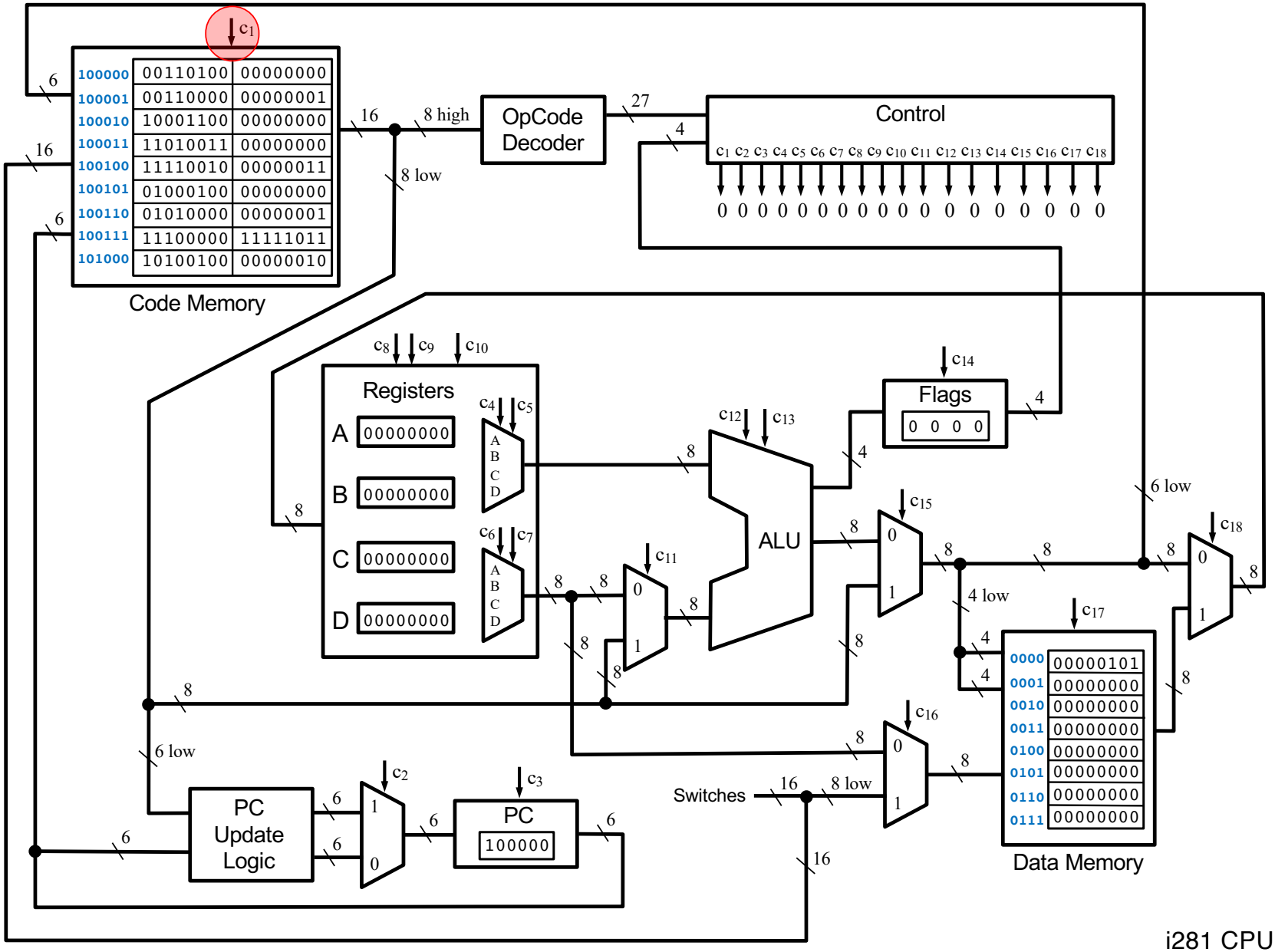
	c_1 IMEM_WRITE_ENABLE	c_2 PROGRAM_COUNTER_MUX	c_3 PROGRAM_COUNTER_WRITE_EN	c_4 REGISTERS_PORT0_SELECT1	c_5 REGISTERS_PORT0_SELECT0	c_6 REGISTERS_PORT1_SELECT1	c_7 REGISTERS_PORT1_SELECT0	c_8 REGISTERS_WRITE_SELECT1	c_9 REGISTERS_WRITE_SELECT0	c_{10} REGISTERS_WRITE_ENABLE	c_{11} ALU_SOURCE_MUX	c_{12} ALU_SELECT1	c_{13} ALU_SELECT0	c_{14} FLAGS_WRITE_ENABLE	c_{15} ALU_RESUT_MUX	c_{16} DMEM_INPUT_MUX	c_{17} DMEM_WRITE_ENABLE	c_{18} REG_WRITEBACK_MUX
NOOP			1															
INPUTC	1		1												1			
INPUTCF	1		1	X1	X0						1	1						
INPUTD			1												1	1	1	
INPUTDF			1	X1	X0						1	1				1	1	
MOVE			1	Y1	Y0			X1	X0	1	1	1						
LOADI/LOADP			1					X1	X0	1					1			
ADD			1	X1	X0	Y1	Y0	X1	X0	1		1		1				
ADDI			1	X1	X0			X1	X0	1	1	1		1				
SUB			1	X1	X0	Y1	Y0	X1	X0	1		1	1	1				
SUBI			1	X1	X0			X1	X0	1	1	1	1	1				
LOAD			1					X1	X0	1					1			1
LOADF			1	Y1	Y0			X1	X0	1	1	1						1
STORE			1			X1	X0								1		1	
STOREF			1	Y1	Y0	X1	X0				1	1					1	
SHIFTL			1	X1	X0			X1	X0	1				1				
SHIFTR			1	X1	X0			X1	X0	1			1	1				
CMP			1	X1	X0	Y1	Y0					1	1	1				
JUMP		1	1															
BRE/BRZ		B1	1															
BRNE/BRNZ		B2	1															
BRG		B3	1															
BRGE		B4	1															

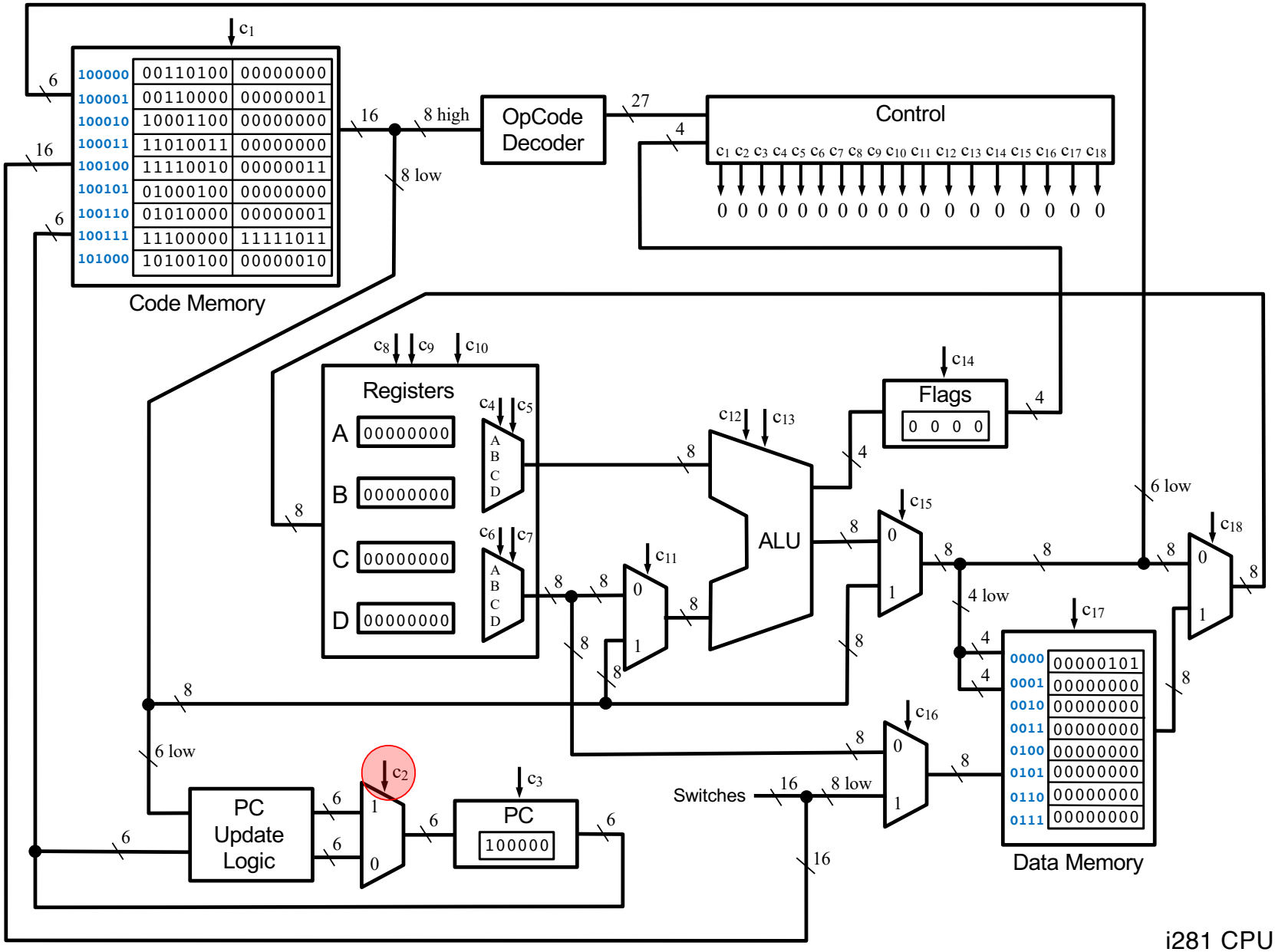
computed using the flags register

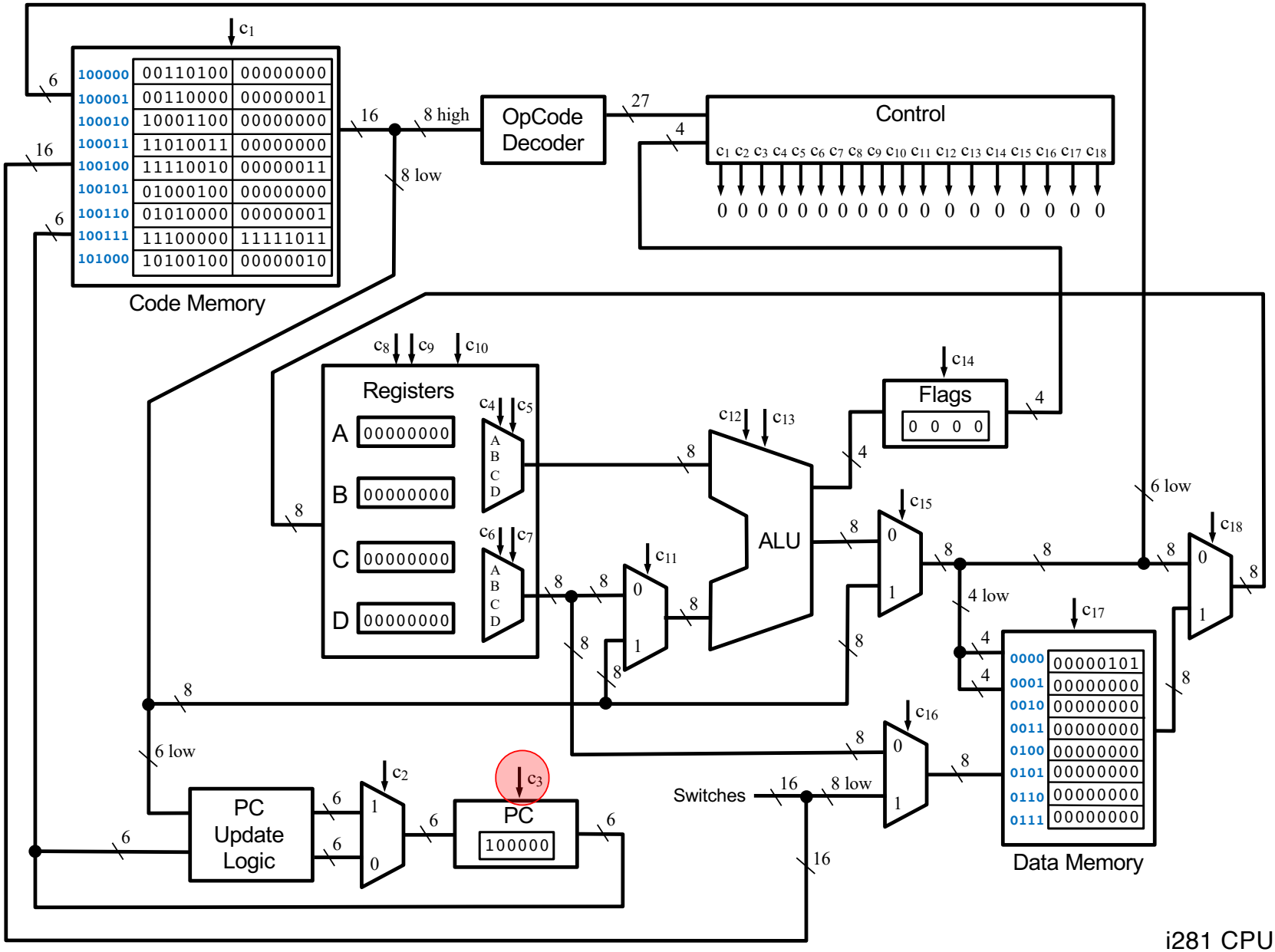
```
B1=  ZF
B2= ~ZF
B3= AND(~ZF, XNOR(NF, OF))
B4= XNOR(NF, OF)
```

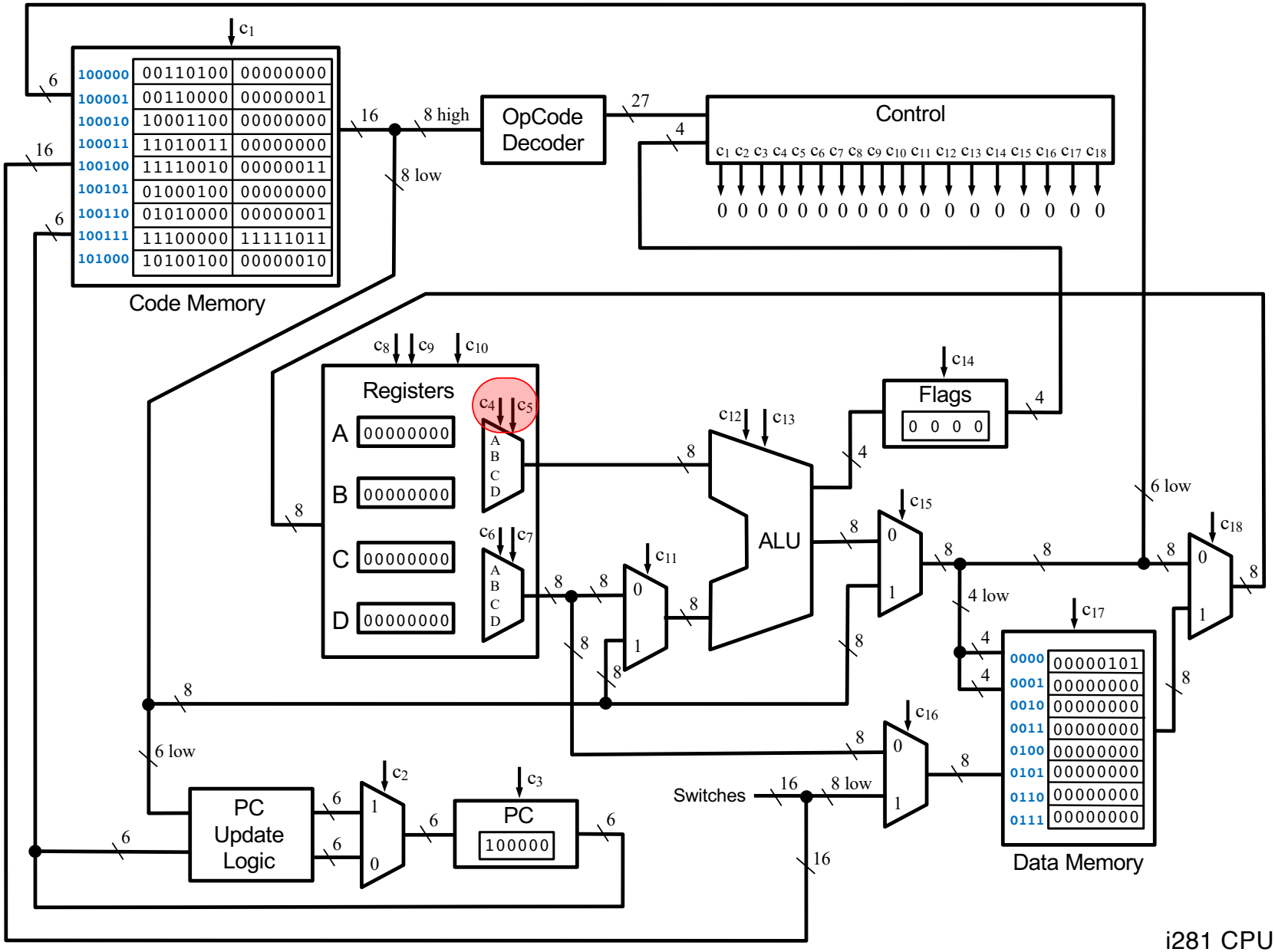
Zero Flag (ZF)
Negative Flag (NF)
Overflow Flag (OF)

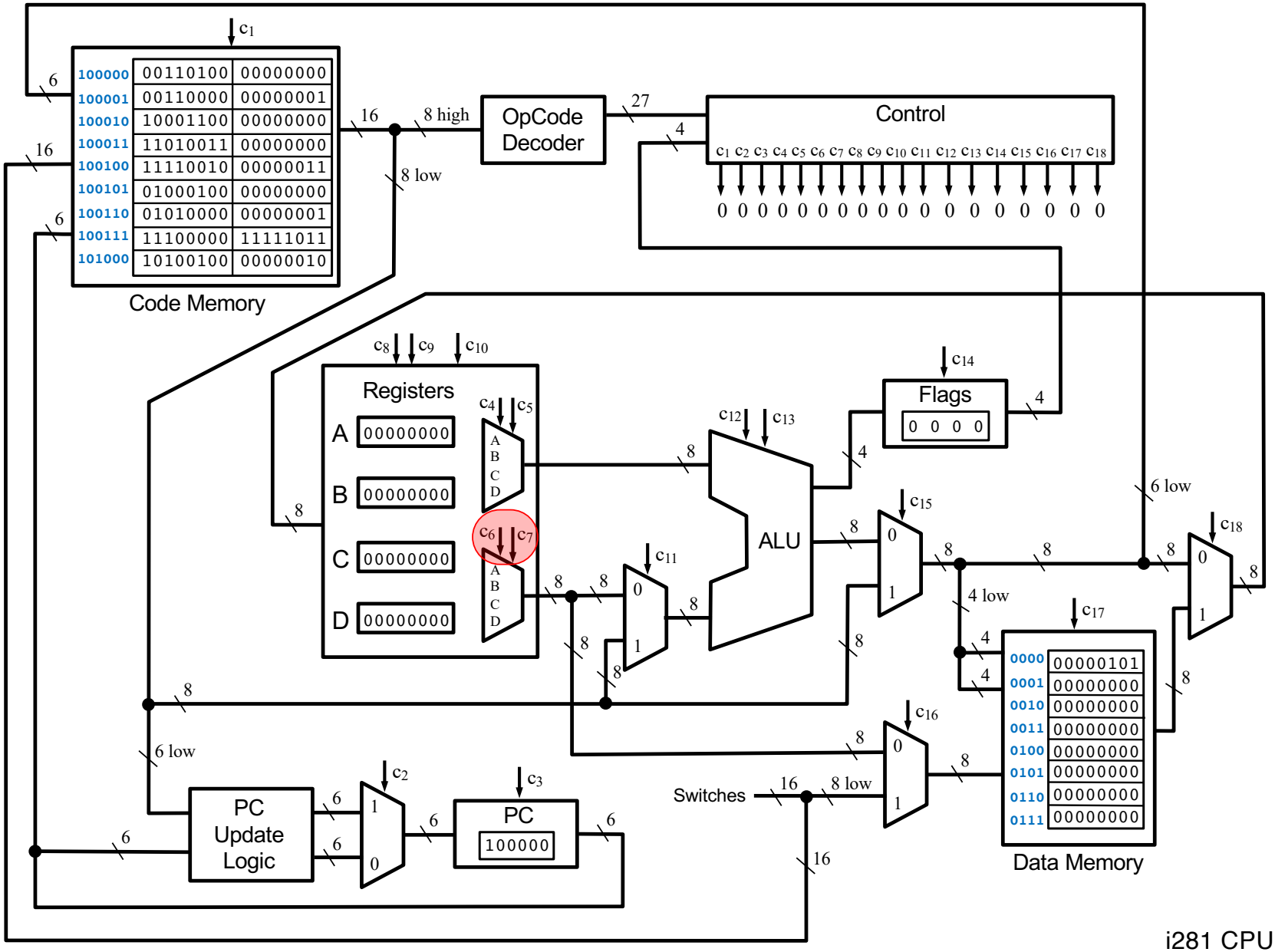
The Control Signals

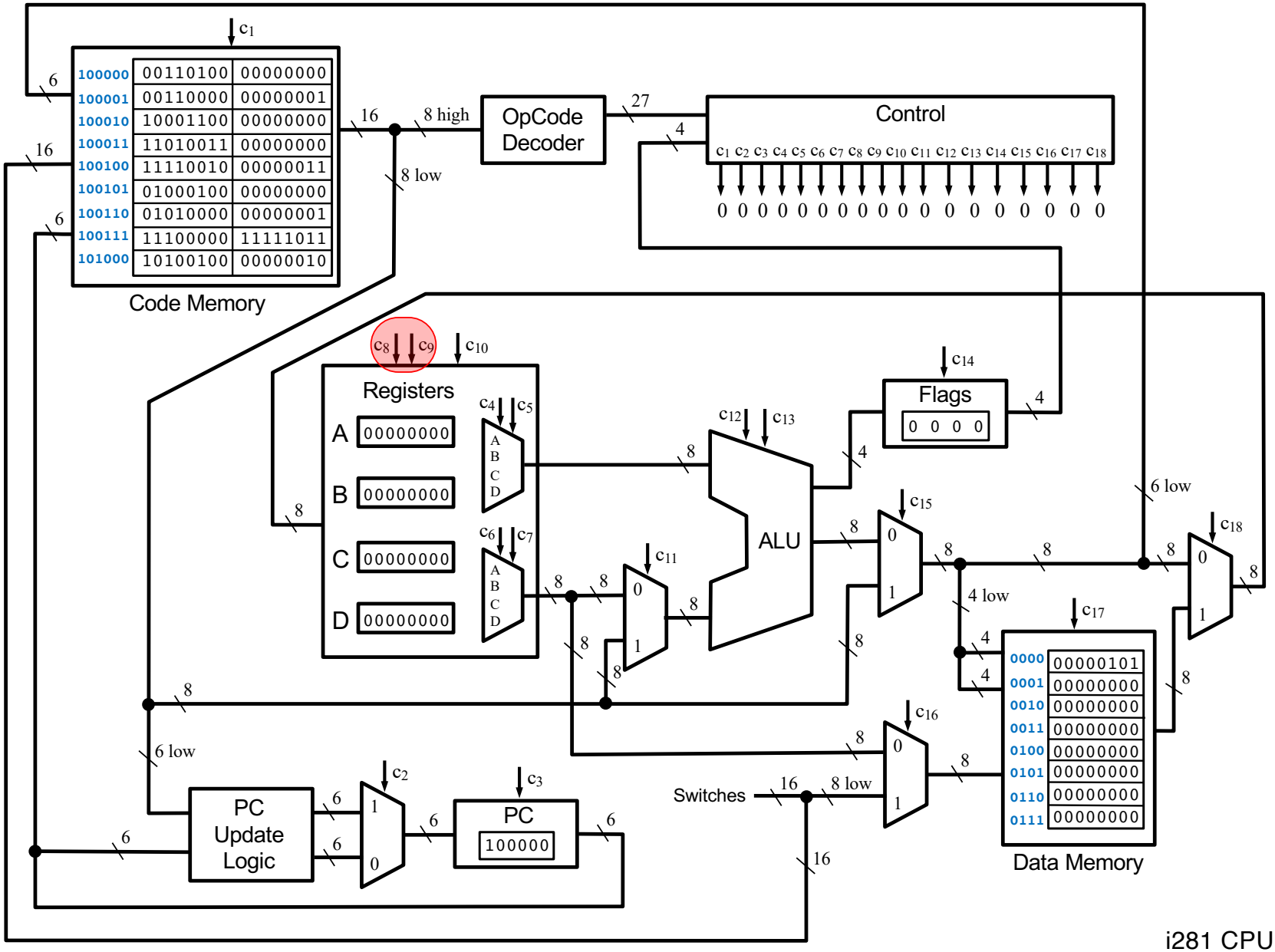

c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c12 c13
ALU
c14
Flags
0 0 0 0
4
c15
c11
c16
c17
c18
6 low
4 low
8 low
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
Switches
16
PC Update Logic
c2
c3
PC
100000
i281 CPU

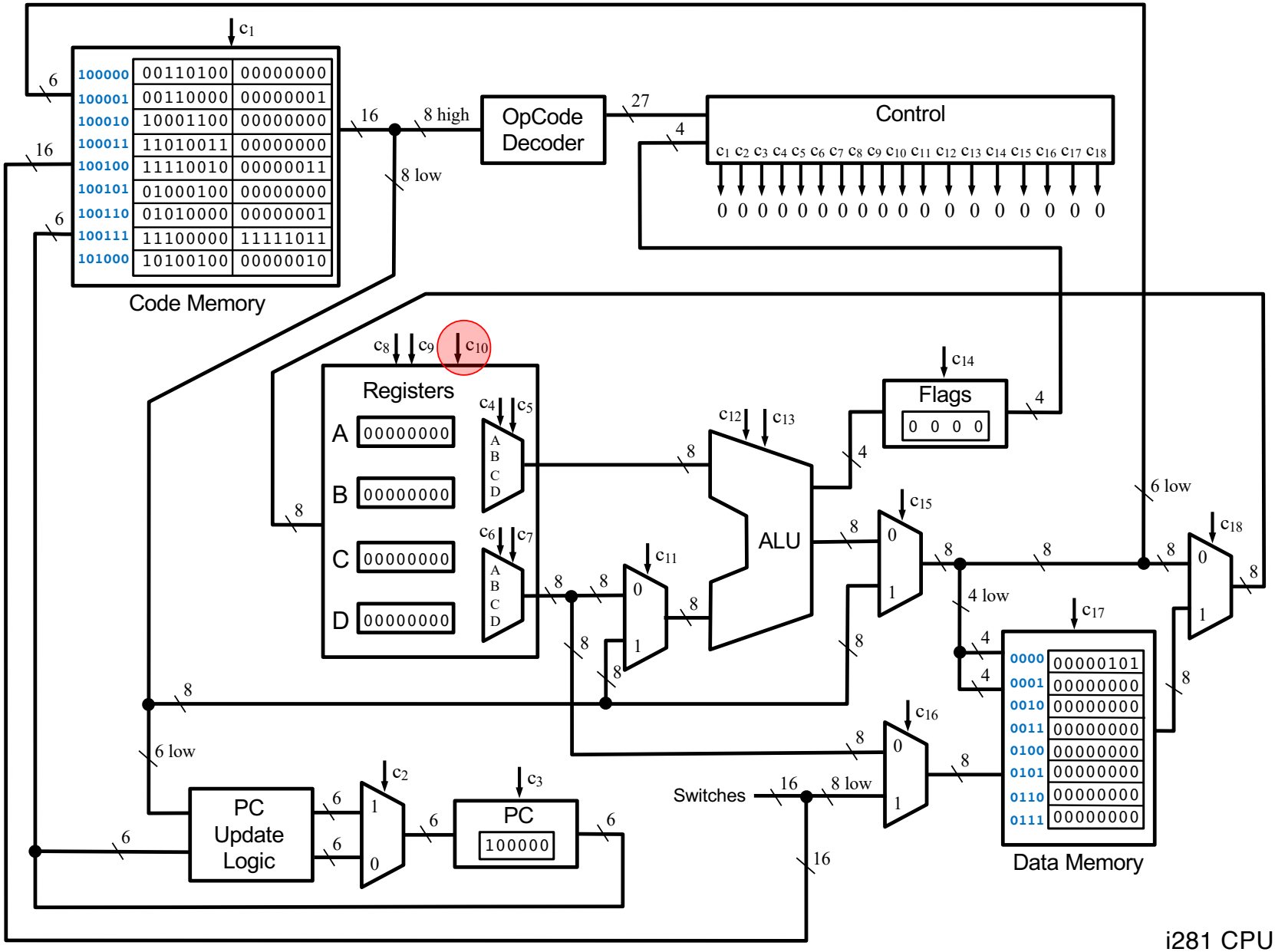

c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c14
Flags
0 0 0 0
4
c12 c13
ALU
c15
c11
c16
c17
c18
6 low
4 low
8 low
Switches
16
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
c2
c3
PC Update Logic
PC
100000
i281 CPU

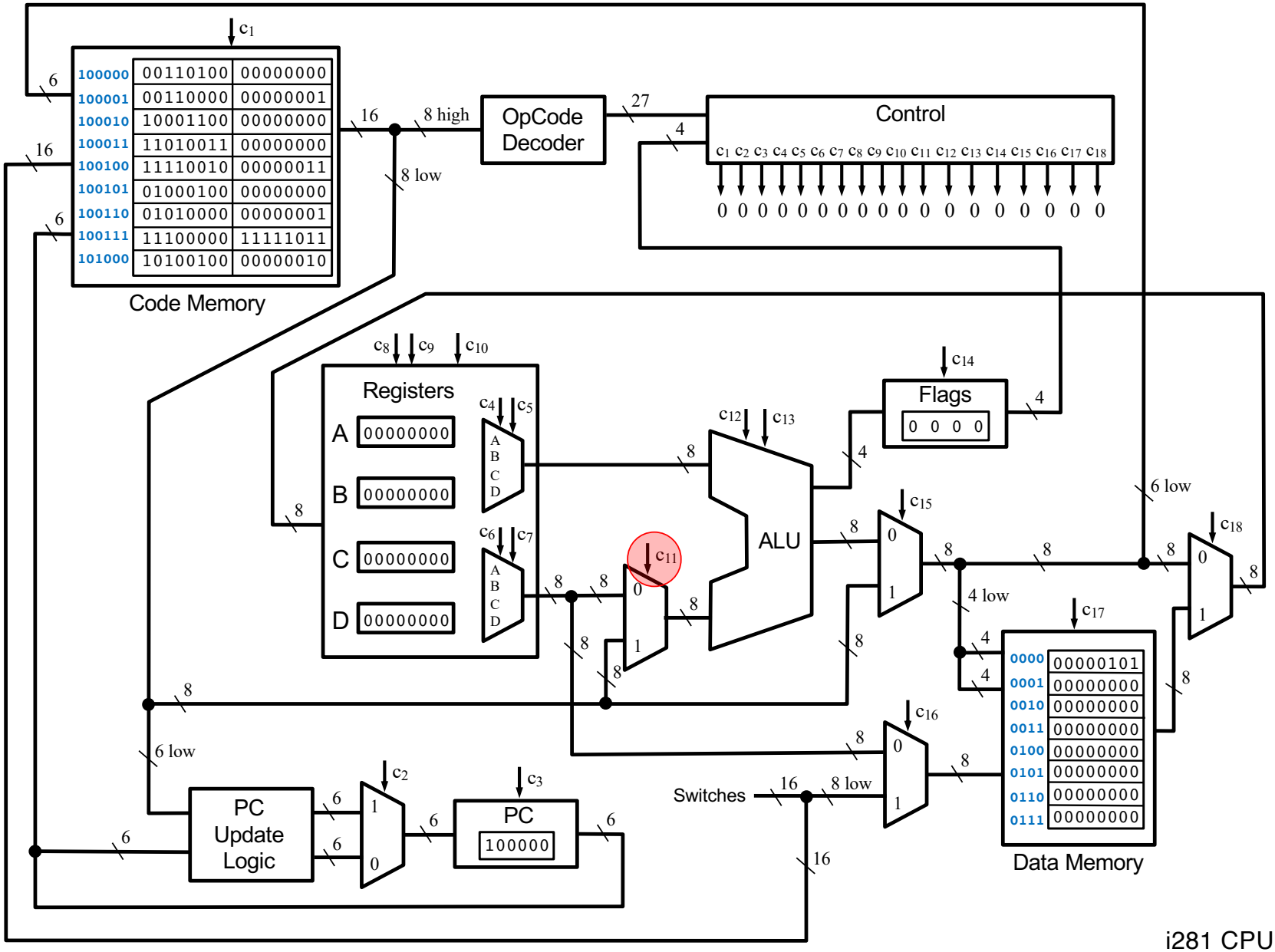

c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c12 c13
ALU
c14
Flags
0 0 0 0
4
c15
c11
c16
c17
c18
6 low
4 low
8 low
Switches
16
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
c2
c3
PC Update Logic
PC
100000
i281 CPU

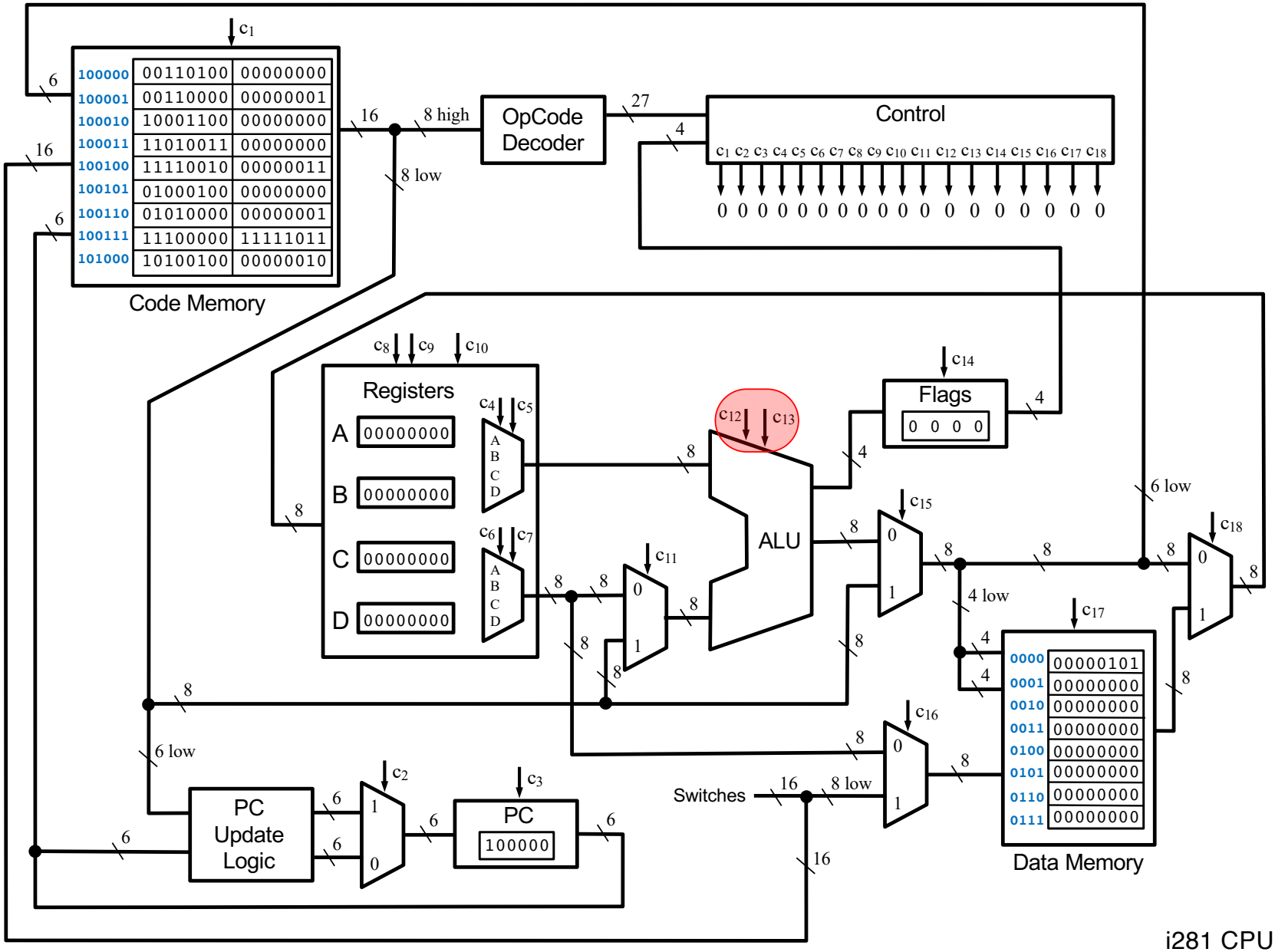

c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c14
Flags
0 0 0 0
4
c12 c13
ALU
c15
c11
c16
c17
c18
6 low
4 low
8 low
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
Switches
16
c2
c3
PC Update Logic
PC
100000
i281 CPU

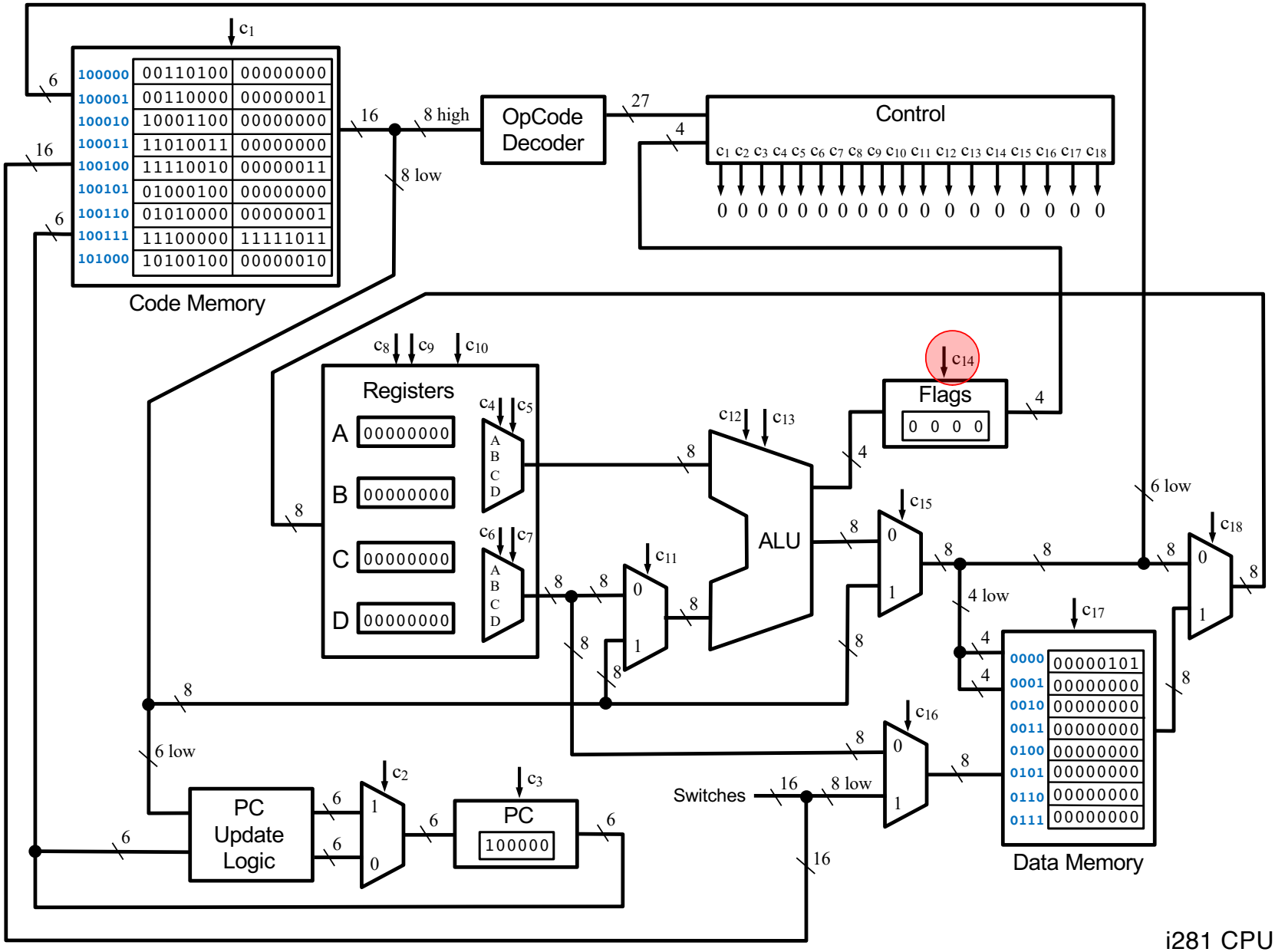

c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c12 c13
ALU
c14
Flags
0 0 0 0
4
c11
c15
c18
6 low
4 low
c17
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
c16
Switches
16
8 low
c2
c3
PC Update Logic
PC
100000
6
6 low
i281 CPU

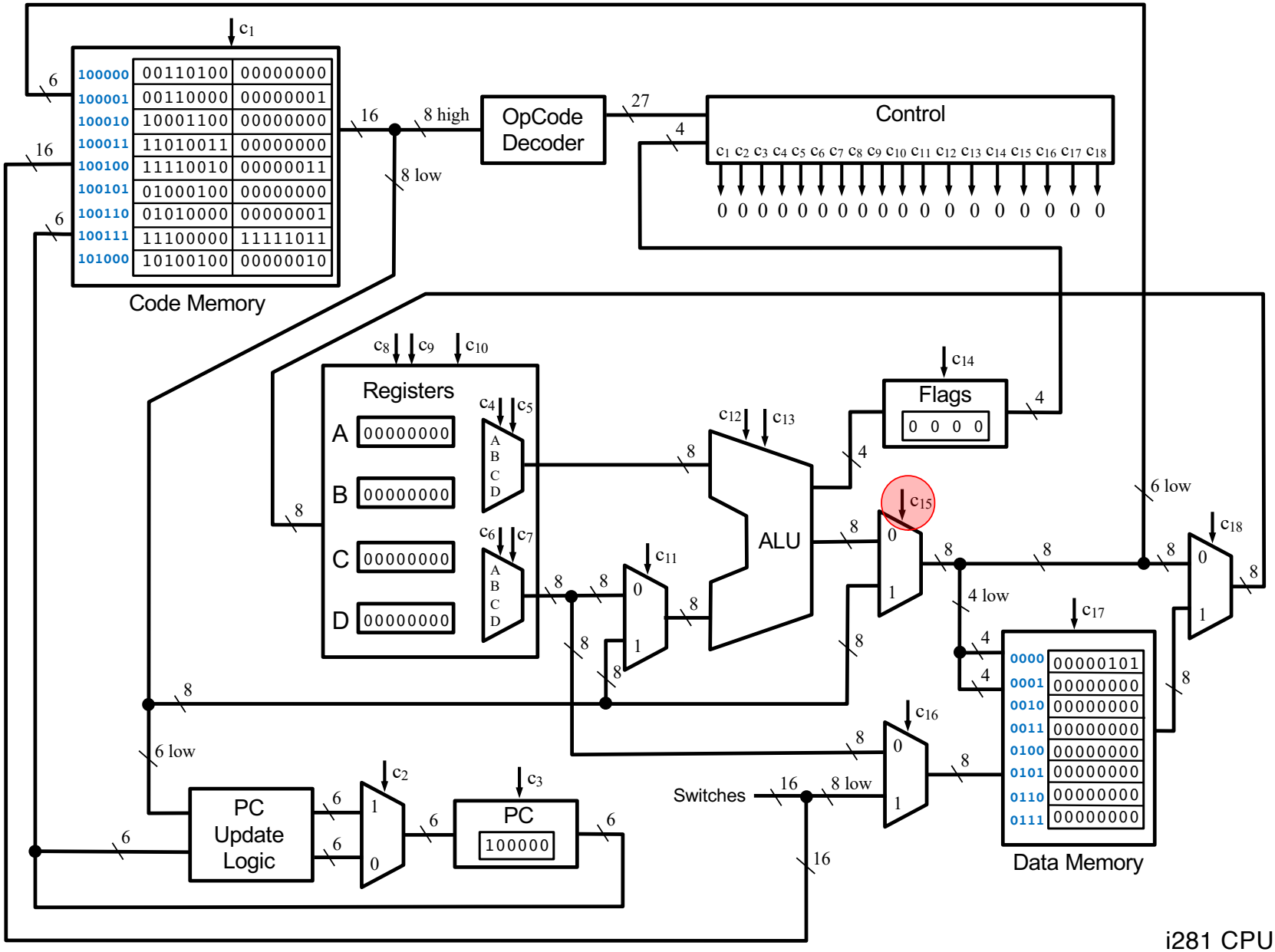

c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c12 c13
ALU
c14
Flags
0 0 0 0
4
c15
c11
c16
c17
c18
6 low
4 low
8 low
Switches
16
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
c2
c3
PC Update Logic
PC
100000
6
i281 CPU

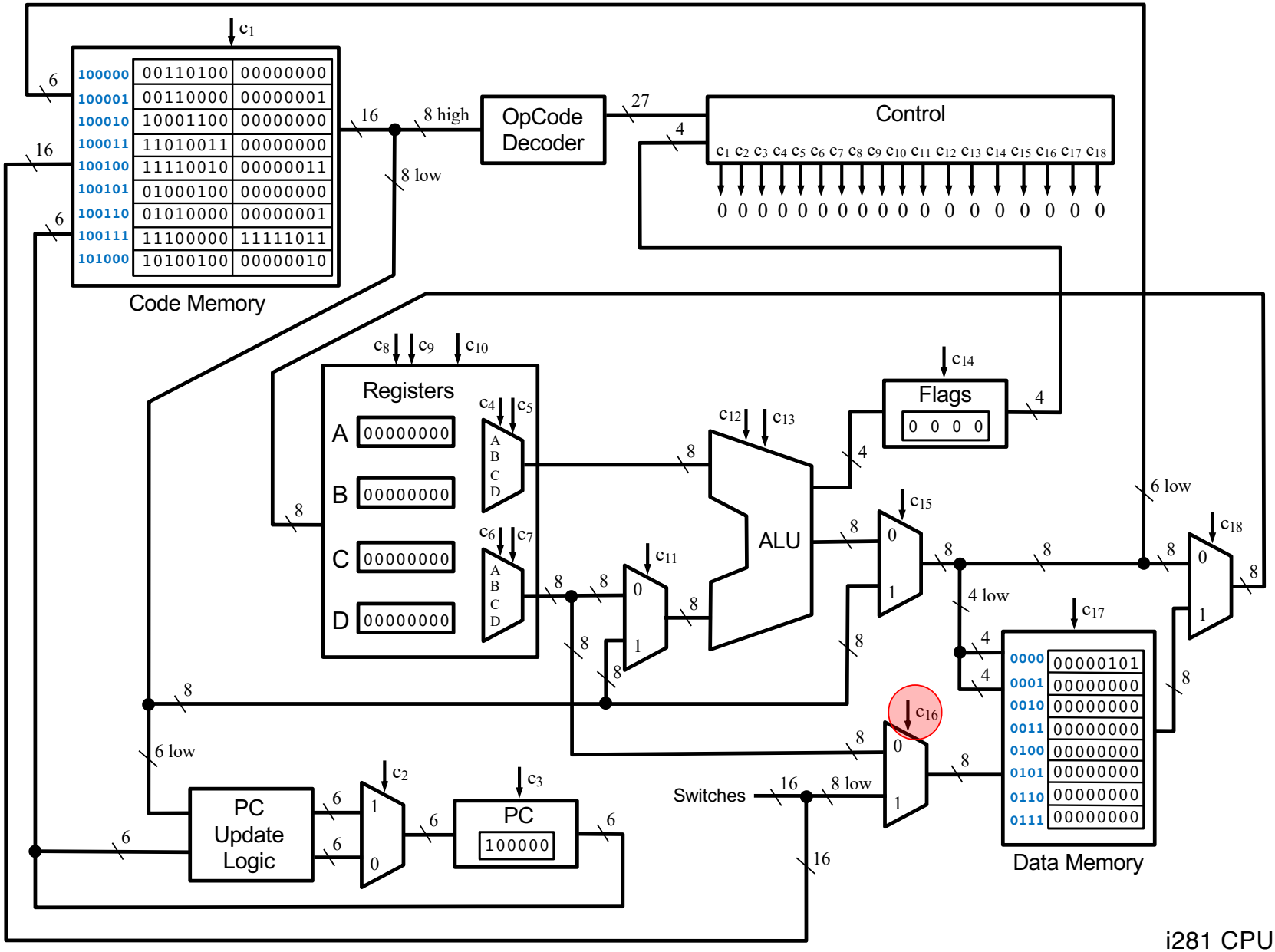

c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c12 c13
ALU
c14
Flags
0 0 0 0
4
c15
c11
c16
c17
c18
6 low
4 low
8 low
Switches
16
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
PC Update Logic
c2
c3
PC
100000
i281 CPU

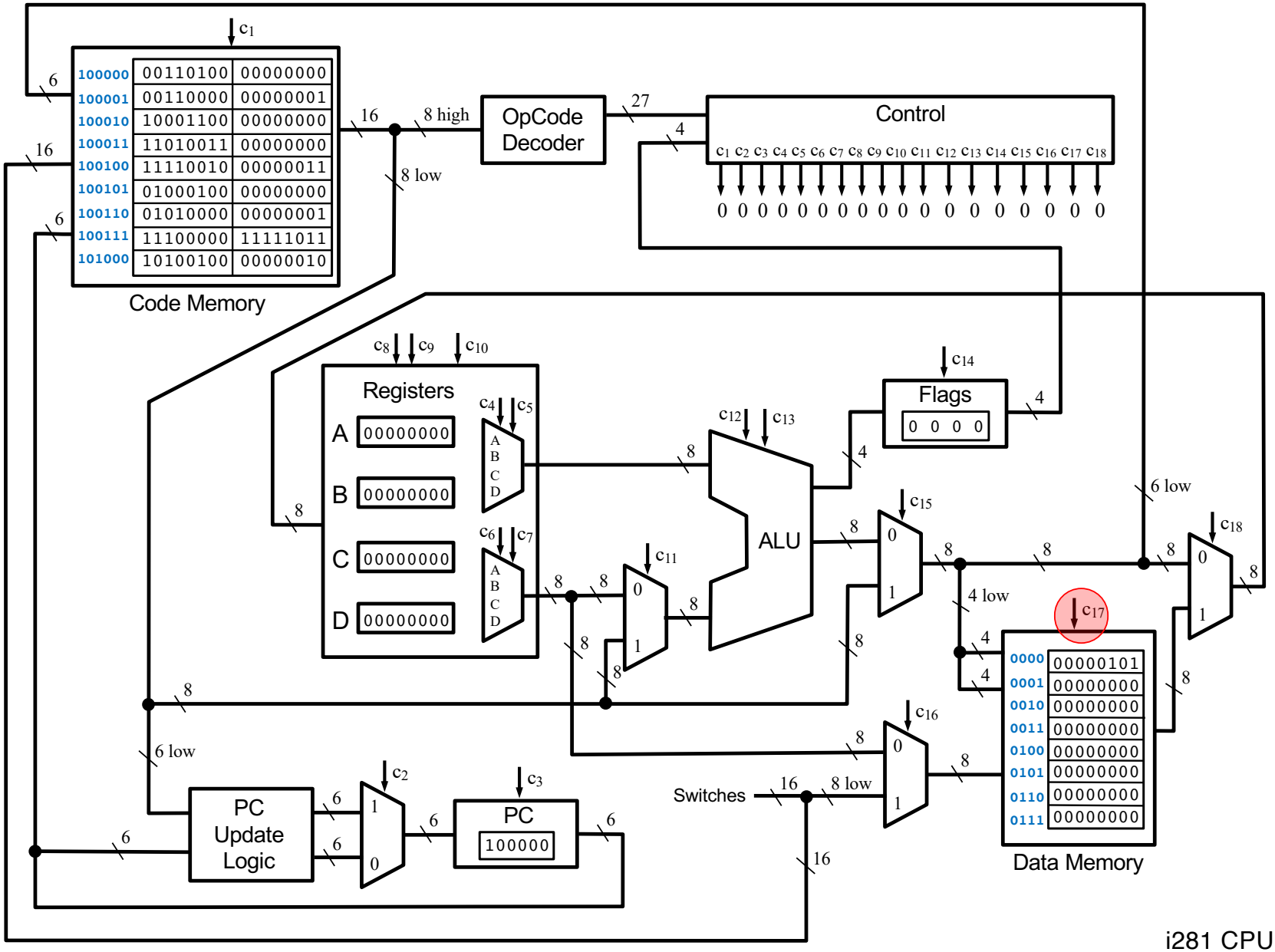

c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c11
c12 c13
ALU
c14
Flags
0 0 0 0
4
c15
c16
c17
c18
6 low
4 low
8 low
Switches
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
c2
c3
PC Update Logic
PC
100000
i281 CPU

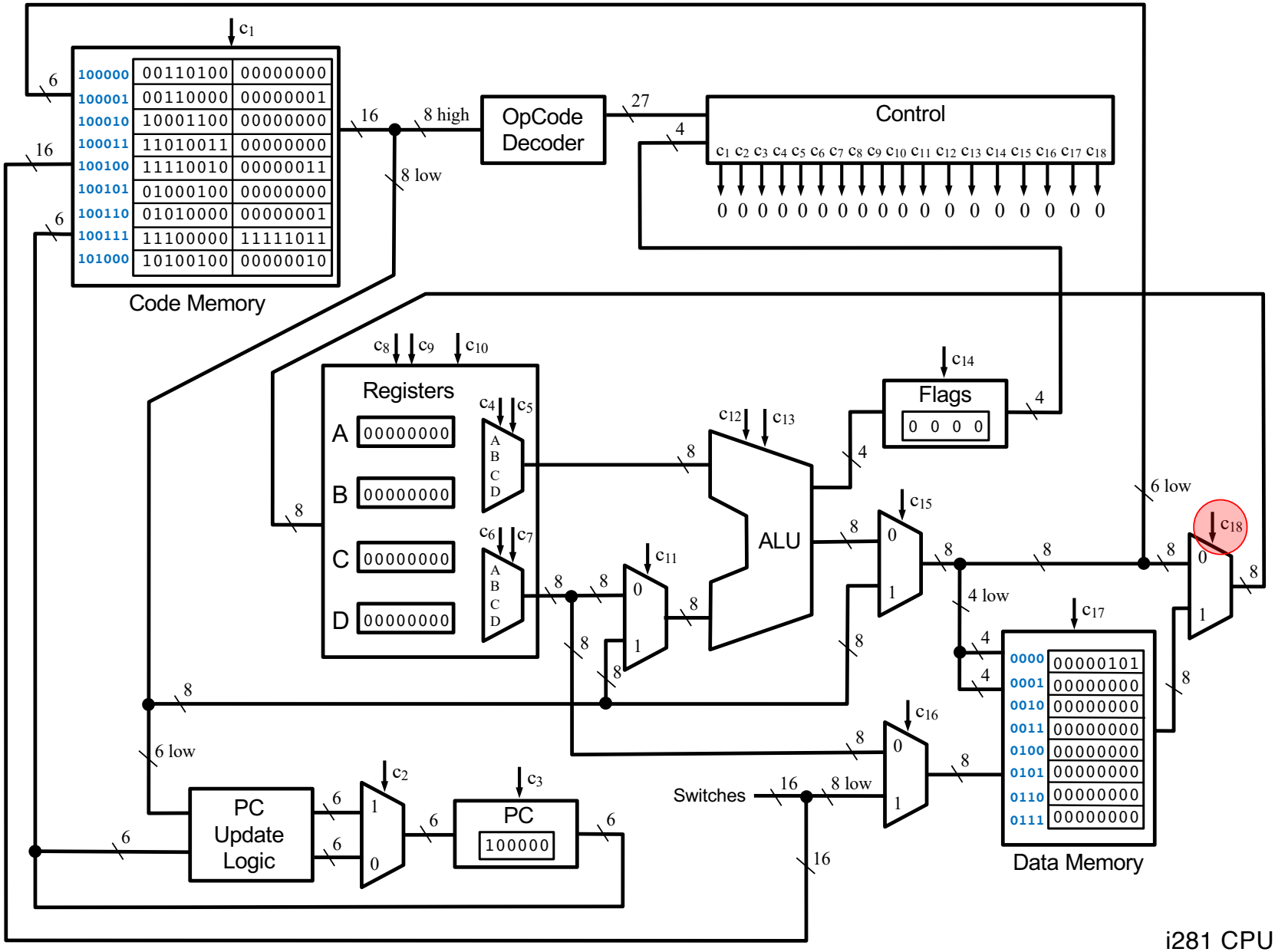

c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c12 c13
ALU
c14
Flags
0 0 0 0
4
c15
c11
c16
c17
c18
6 low
4 low
8 low
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
Switches
16
c2
c3
PC Update Logic
PC
100000
6
i281 CPU

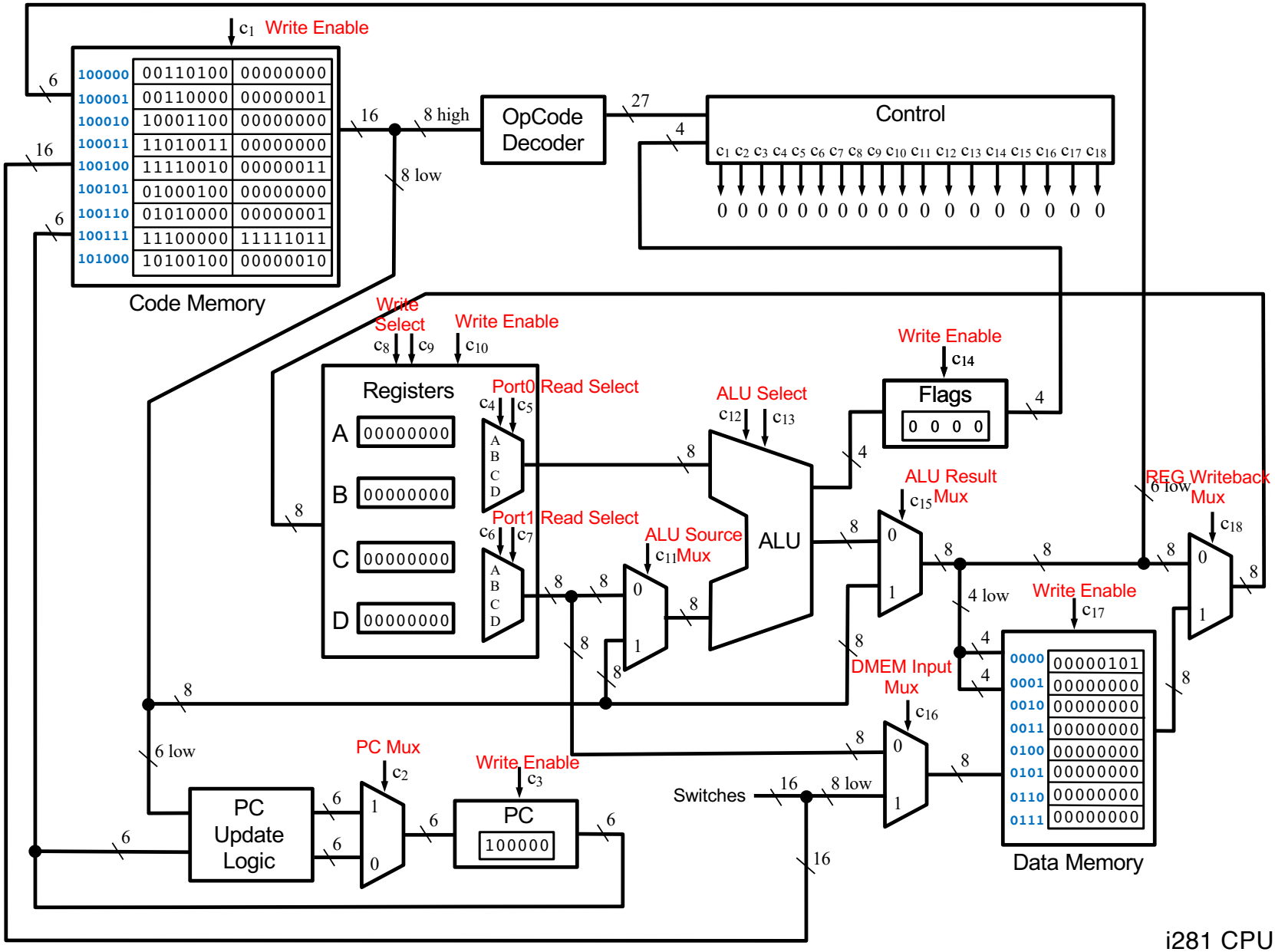

c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c12 c13
ALU
c14
Flags
0 0 0 0
4
c15
c11
c16
c17
c18
6 low
4 low
8 low
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
Switches
16
c2
c3
PC Update Logic
PC
100000
6
i281 CPU

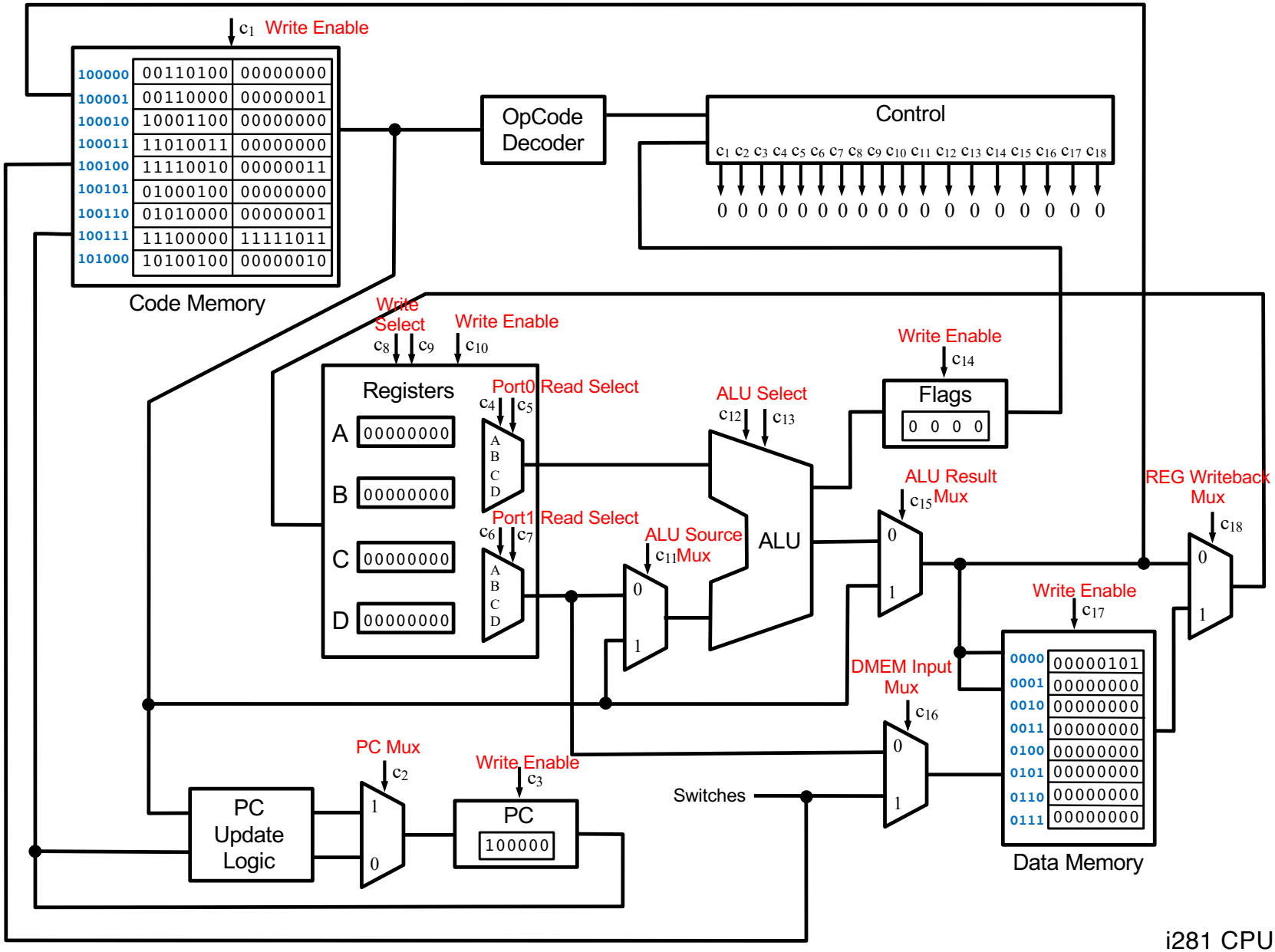

c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c12 c13
ALU
c14
Flags
0 0 0 0
4
c15
c11
c16
c17
c18
6 low
4 low
8 low
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
Switches
16
c2
c3
PC Update Logic
PC
100000
i281 CPU


c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c12 c13
ALU
c14
Flags
0 0 0 0
4
c15
c11
c16
c17
c18
6 low
4 low
8 low
Switches
16
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
c2
c3
PC Update Logic
PC
100000
i281 CPU

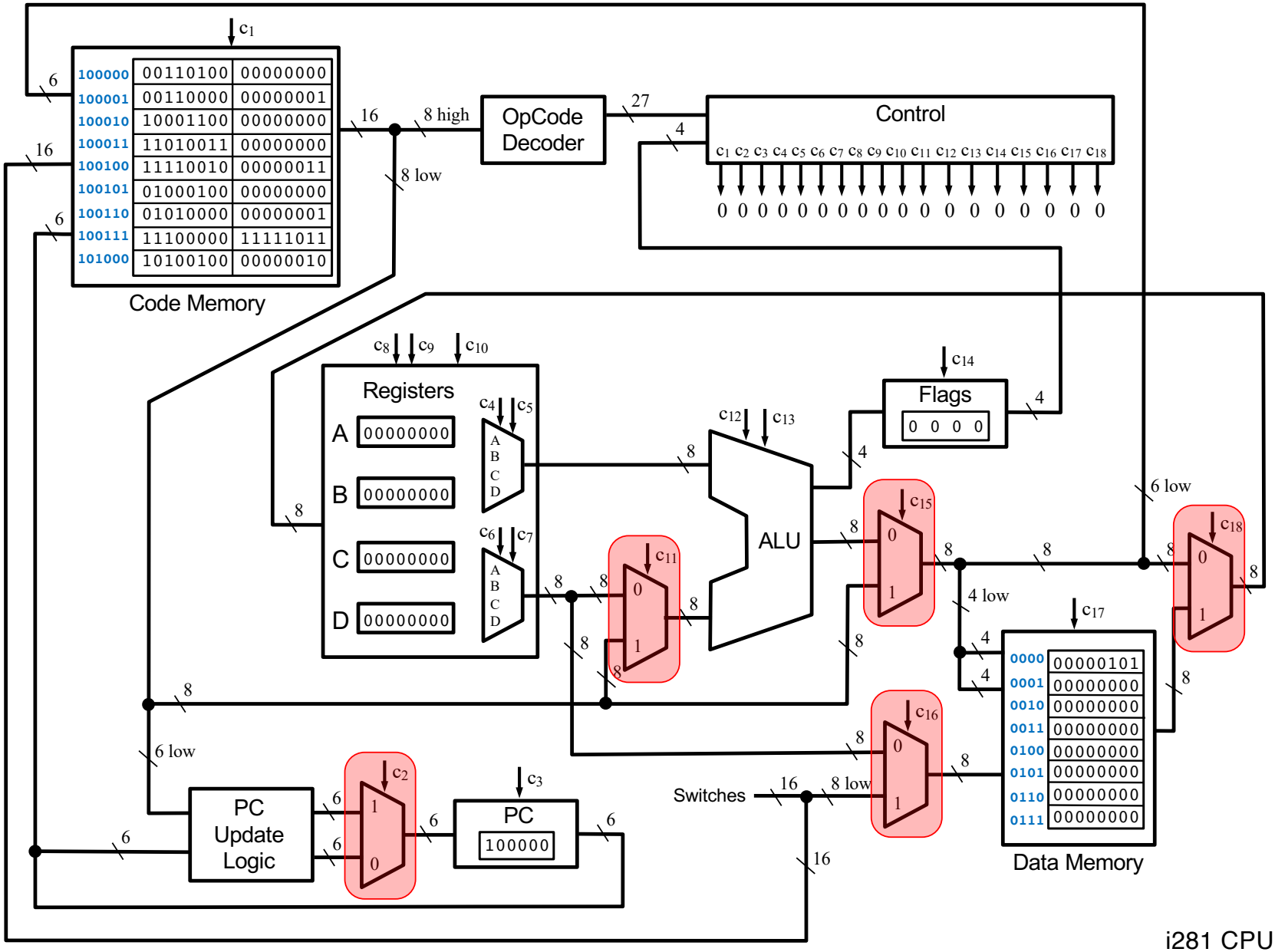

c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c14
Flags
0 0 0 0
4
c12 c13
ALU
c15
c11
c16
c18
6 low
4 low
c17
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
Switches
16
8 low
PC Update Logic
c2
c3
PC
100000
i281 CPU

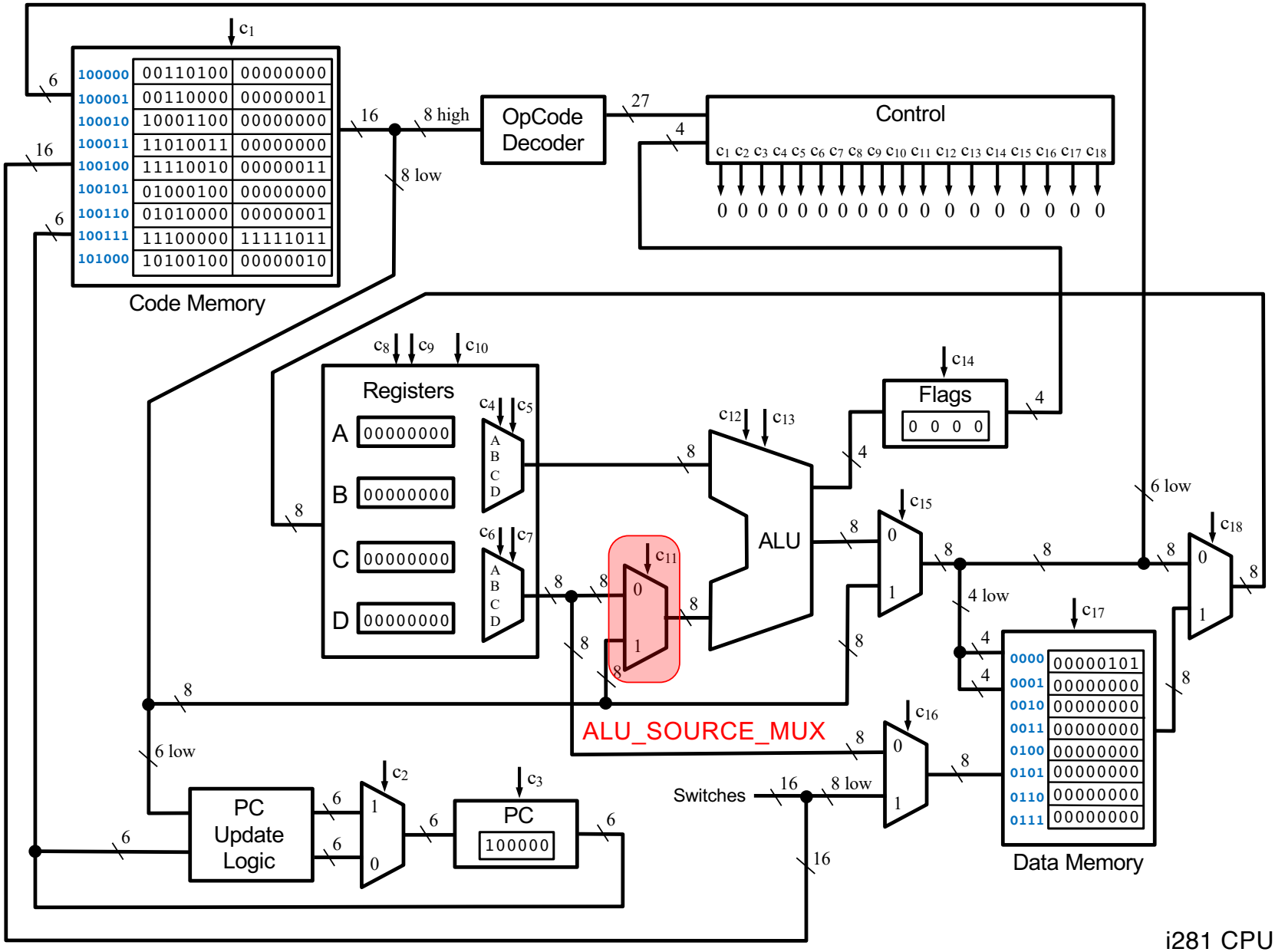

c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c12 c13
ALU
c14
Flags
0 0 0 0
4
c15
c11
c16
c17
c18
6 low
4 low
8 low
Switches
16
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
c2
c3
PC Update Logic
PC
100000
i281 CPU


c1 Write Enable
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Write Select
c8 c9
Write Enable
c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
Port0 Read Select
c4 c5
Port1 Read Select
c6 c7
ALU Select
c12 c13
ALU
ALU Source
c11 Mux
Write Enable
c14
Flags
0 0 0 0
ALU Result
c15 Mux
REG Writeback
Mux
c18
6 low
4 low
Write Enable
c17
DMEM Input
Mux
c16
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
Switches
PC Mux
c2
Write Enable
c3
PC
100000
PC
Update
Logic
i281 CPU

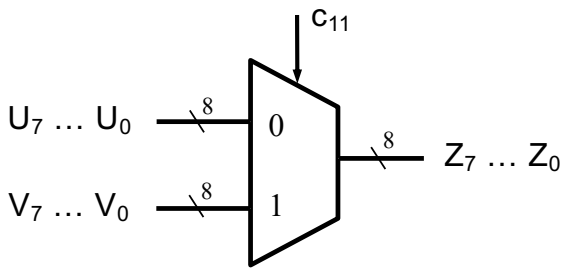

c1 Write Enable
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
OpCode Decoder
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Write Select
c8 c9
Write Enable
c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
Port0 Read Select
c4 c5
Port1 Read Select
c6 c7
A B C D
ALU Select
c12 c13
ALU
ALU Source
c11 Mux
Write Enable
c14
Flags
0 0 0 0
ALU Result
c15 Mux
REG Writeback
Mux
c18
Write Enable
c17
DMEM Input
Mux
c16
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
PC Mux
c2
Write Enable
c3
PC
100000
PC
Update
Logic
Switches
i281 CPU

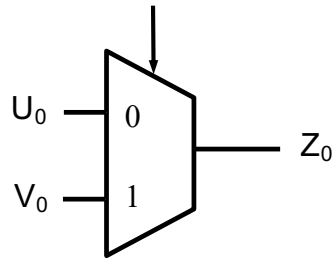
The Five Bus Multiplexers

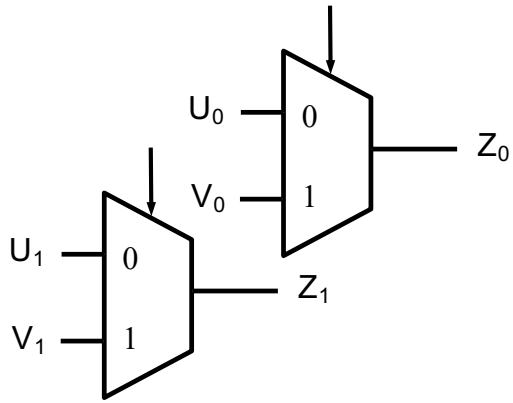

c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c14
Flags
0 0 0 0
4
c12 c13
ALU
c15
c11
c16
c17
c18
6 low
4 low
8 low
Switches
16
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
PC Update Logic
c2
c3
PC
100000
i281 CPU

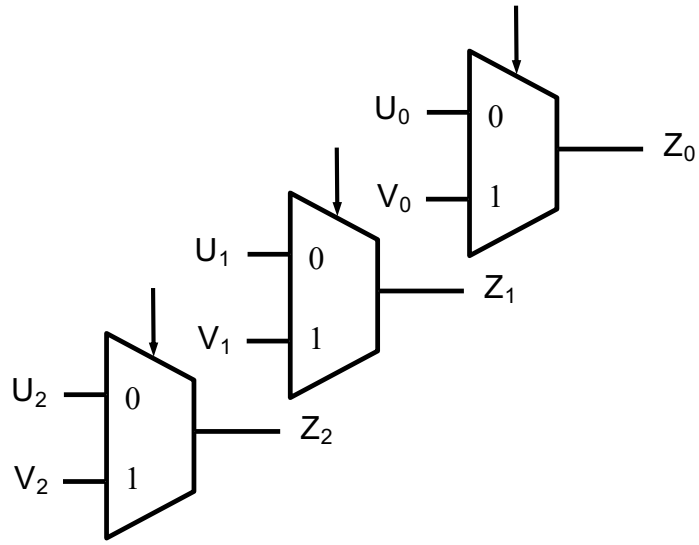

c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
c11
c12 c13
ALU
c14
Flags
0 0 0 0
c15
c16
c17
c18
6 low
4 low
8 low
ALU_SOURCE_MUX
Switches
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
c2
c3
PC Update Logic
PC
100000
i281 CPU

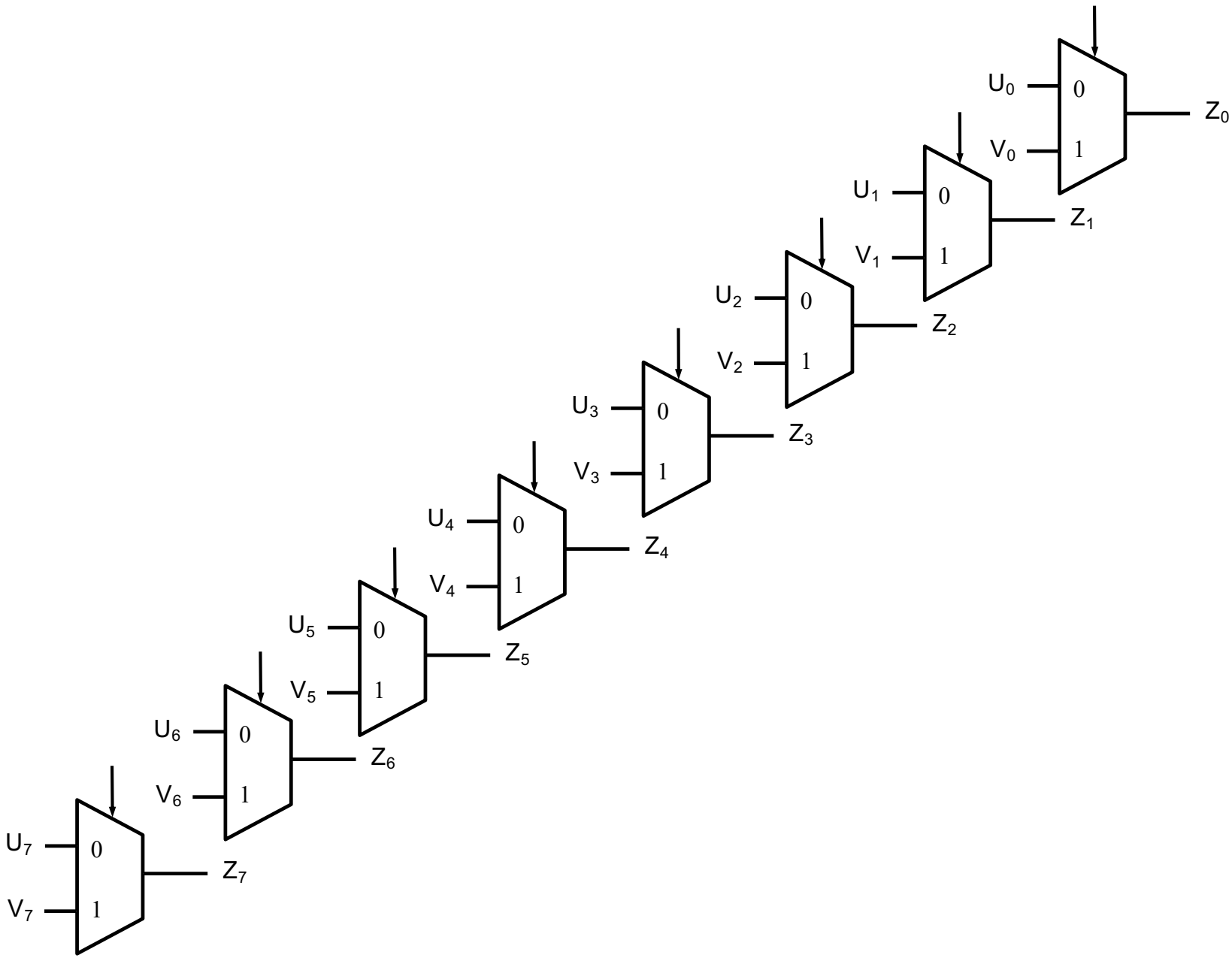
2-to-1 Bus Multiplexer (with 8-bit lines)

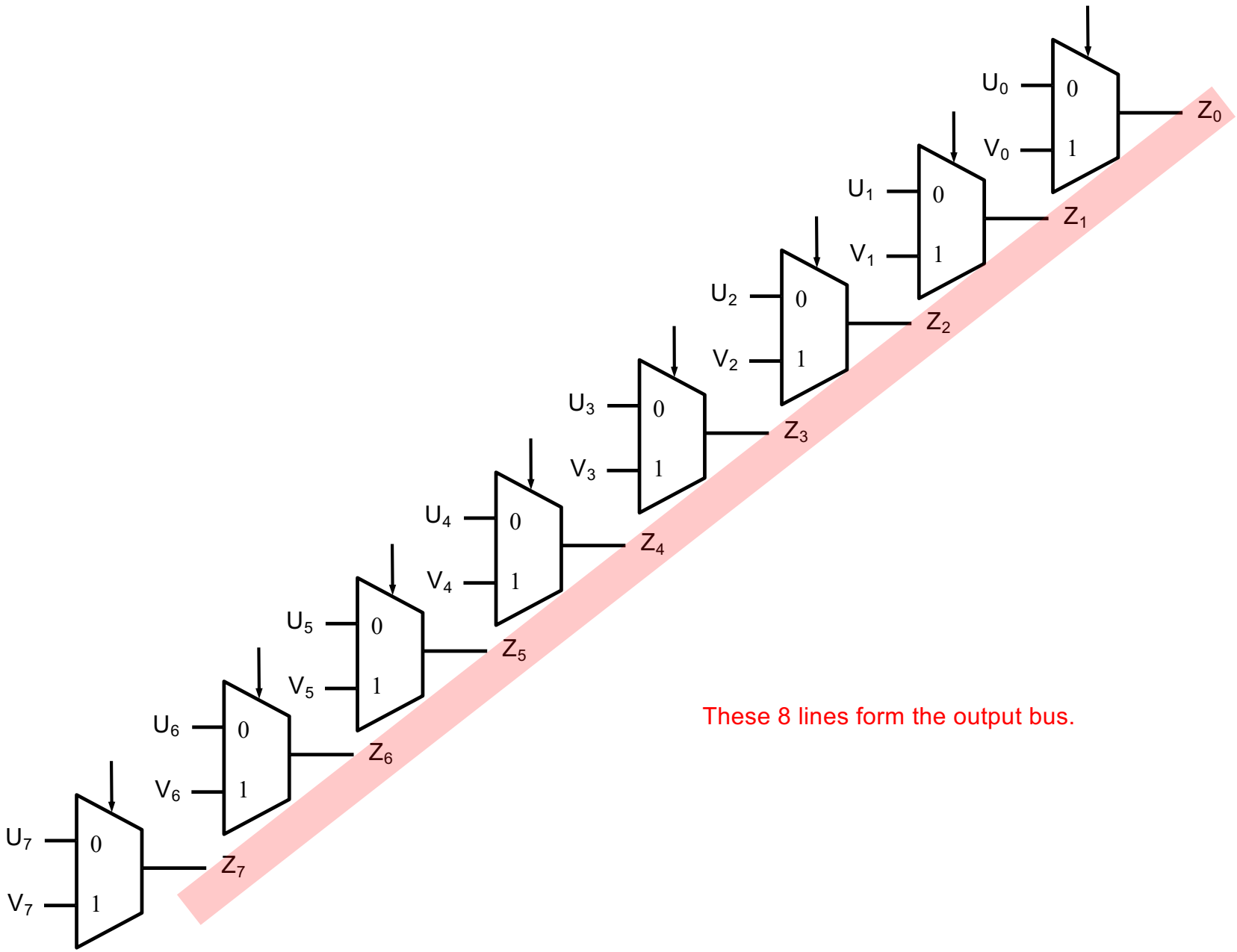


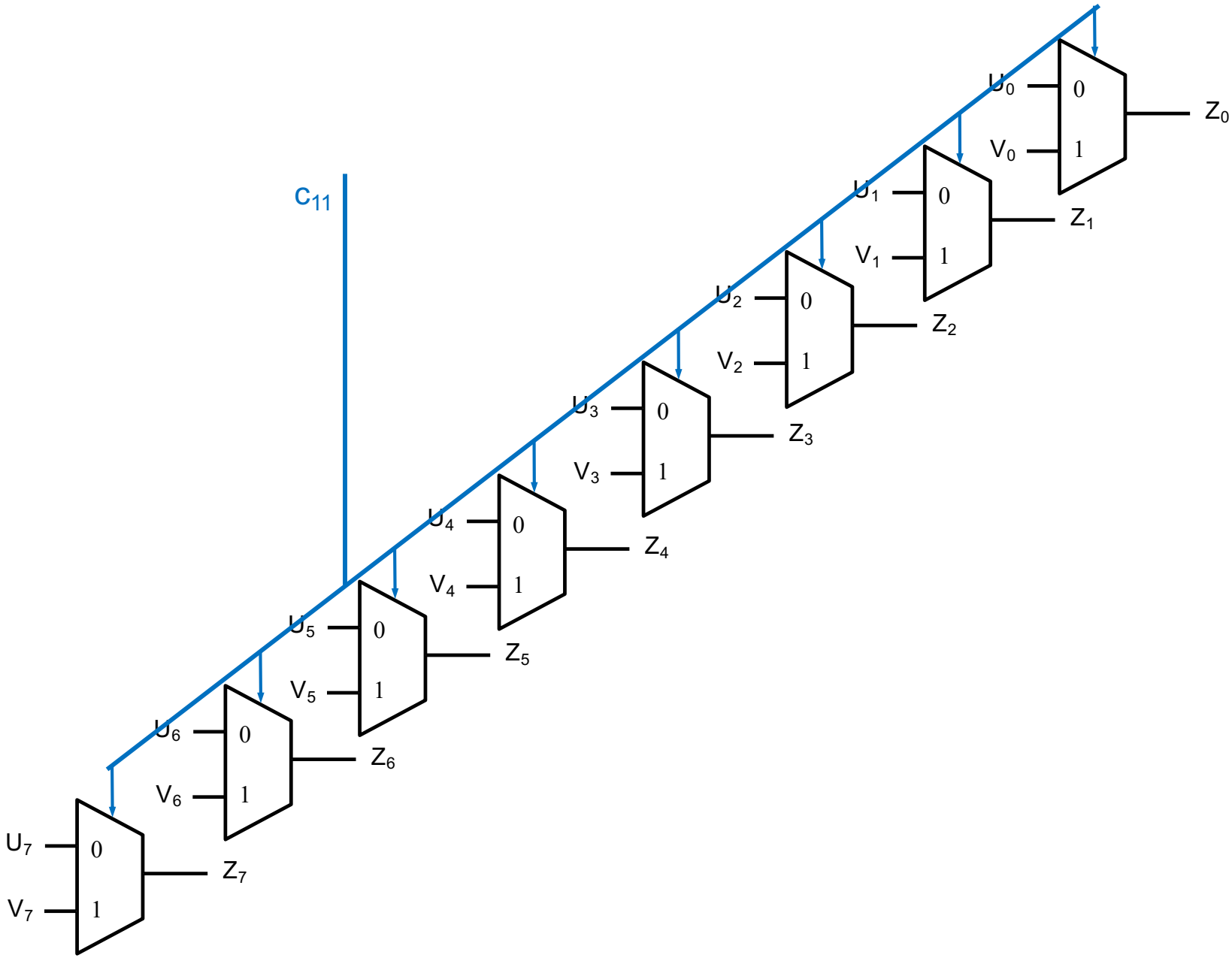

U0
V0
0
1
Z0

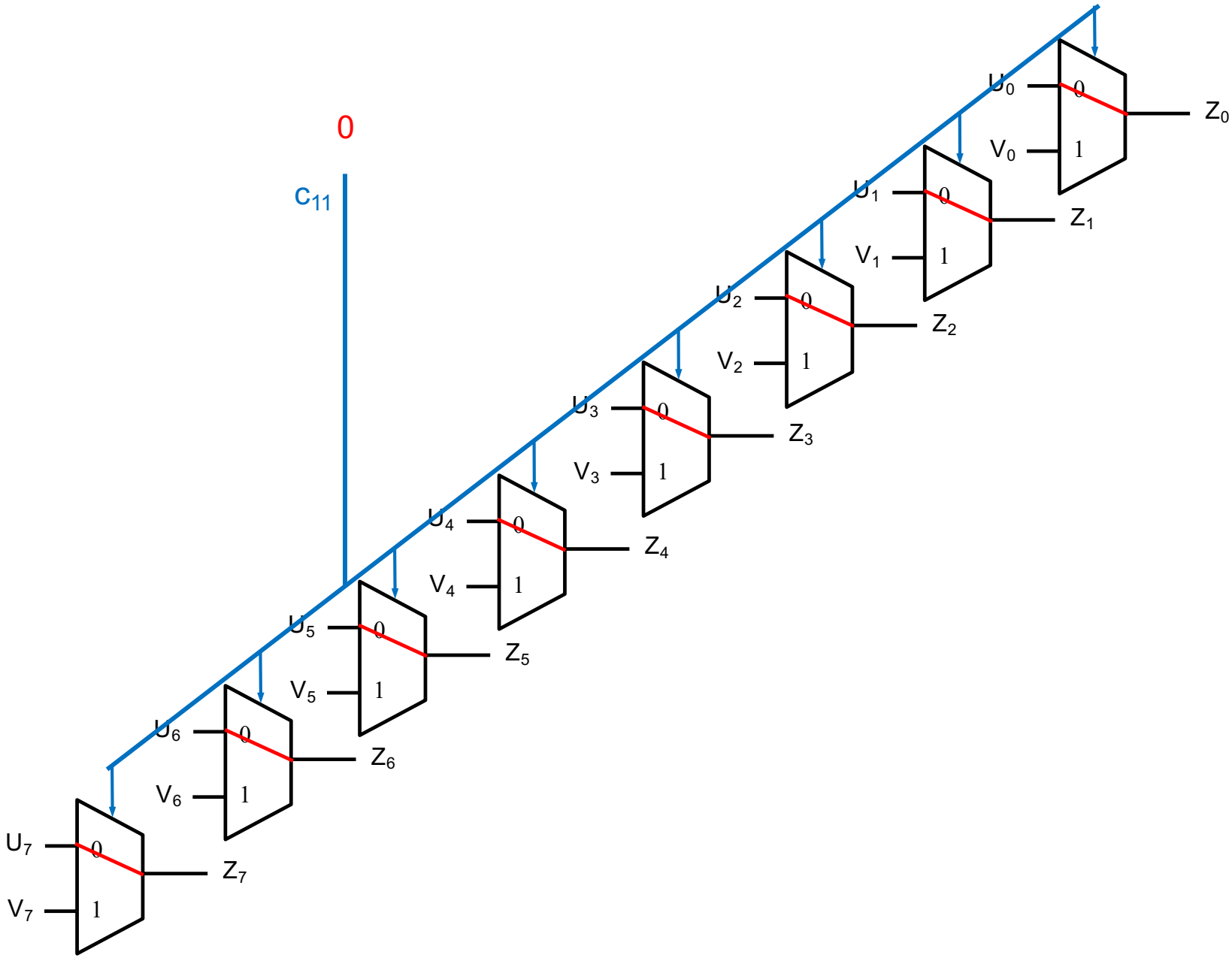

U0
0
Z0
V0
1
U1
0
Z1
V1
1

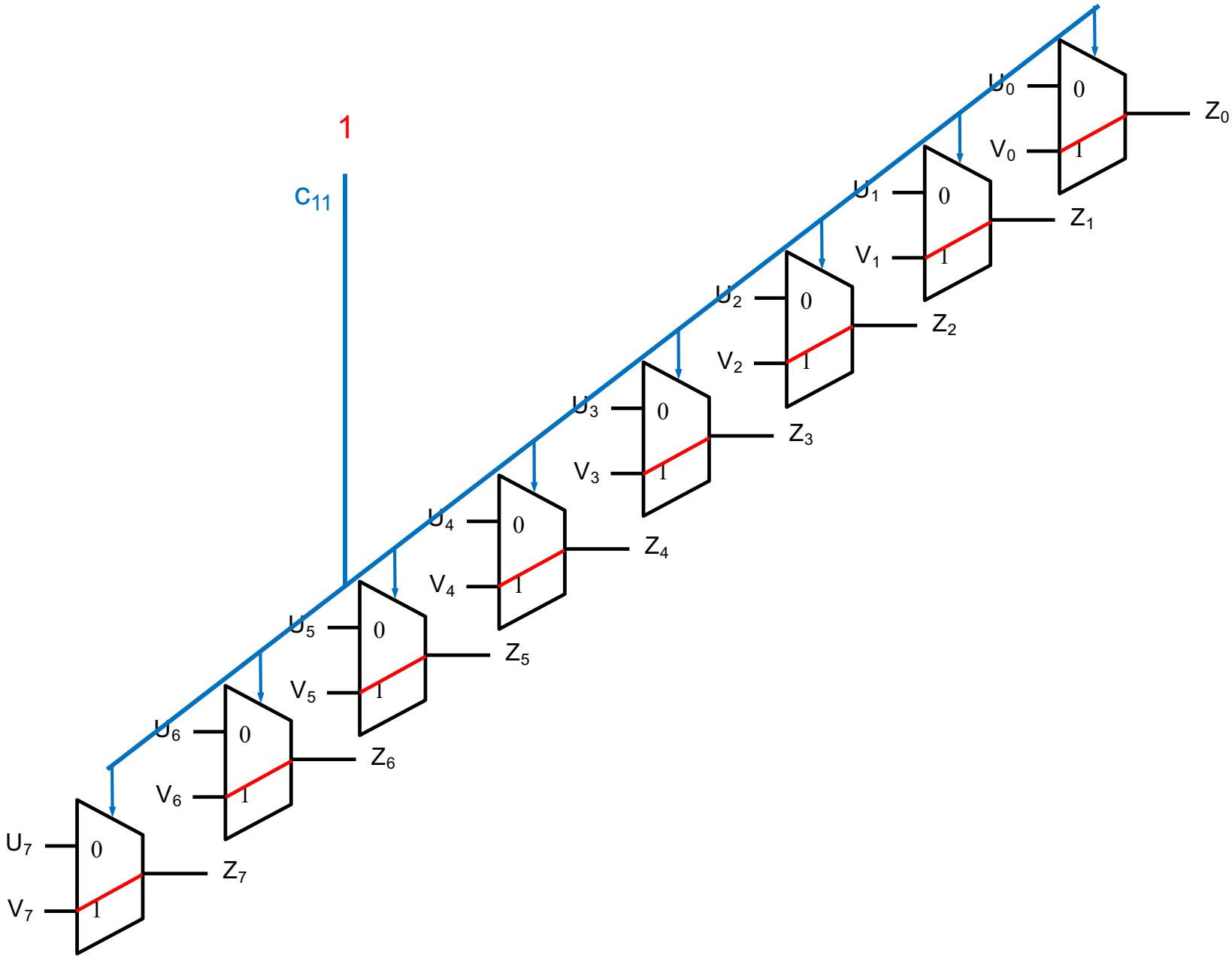

U0
0
Z0
V0
1
U1
0
Z1
V1
1
U2
0
Z2
V2
1

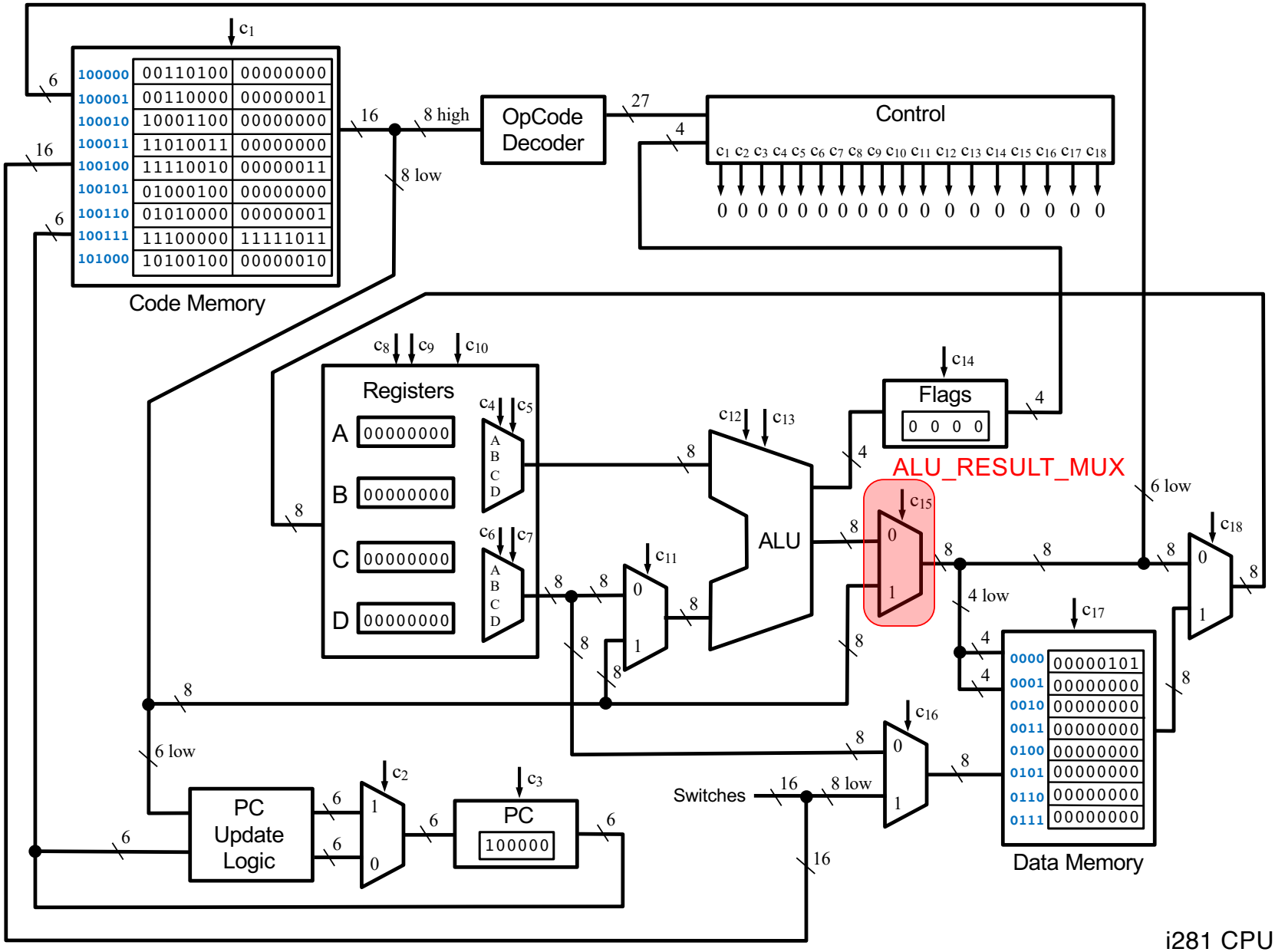

U0
V0
0
1
Z0
U1
V1
0
1
Z1
U2
V2
0
1
Z2
U3
V3
0
1
Z3
U4
V4
0
1
Z4
U5
V5
0
1
Z5
U6
V6
0
1
Z6
U7
V7
0
1
Z7

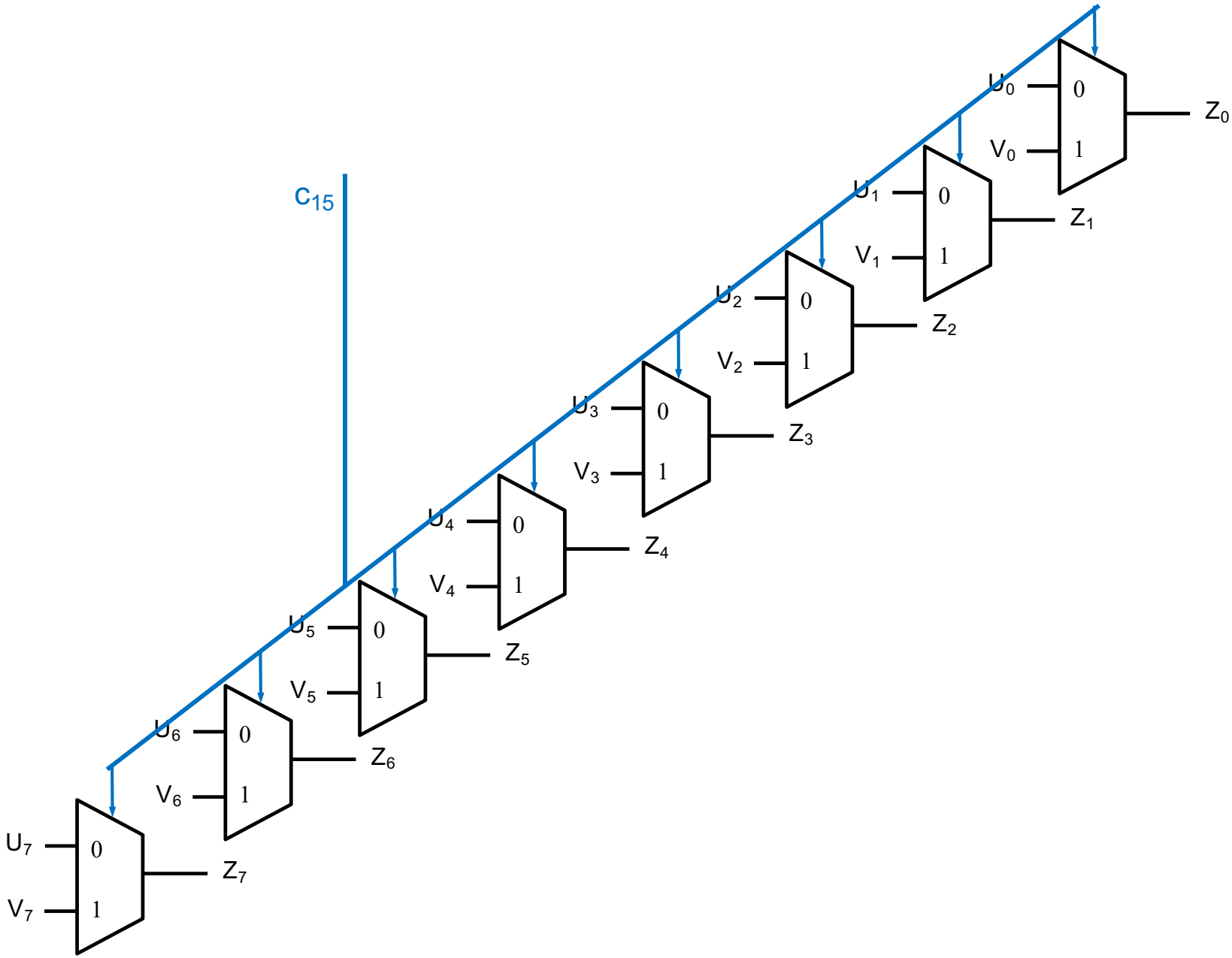

U0
V0
0
1
Z0
U1
V1
Z1
U2
V2
Z2
U3
V3
Z3
U4
V4
Z4
U5
V5
Z5
U6
V6
Z6
U7
V7
Z7
These 8 lines form the output bus.

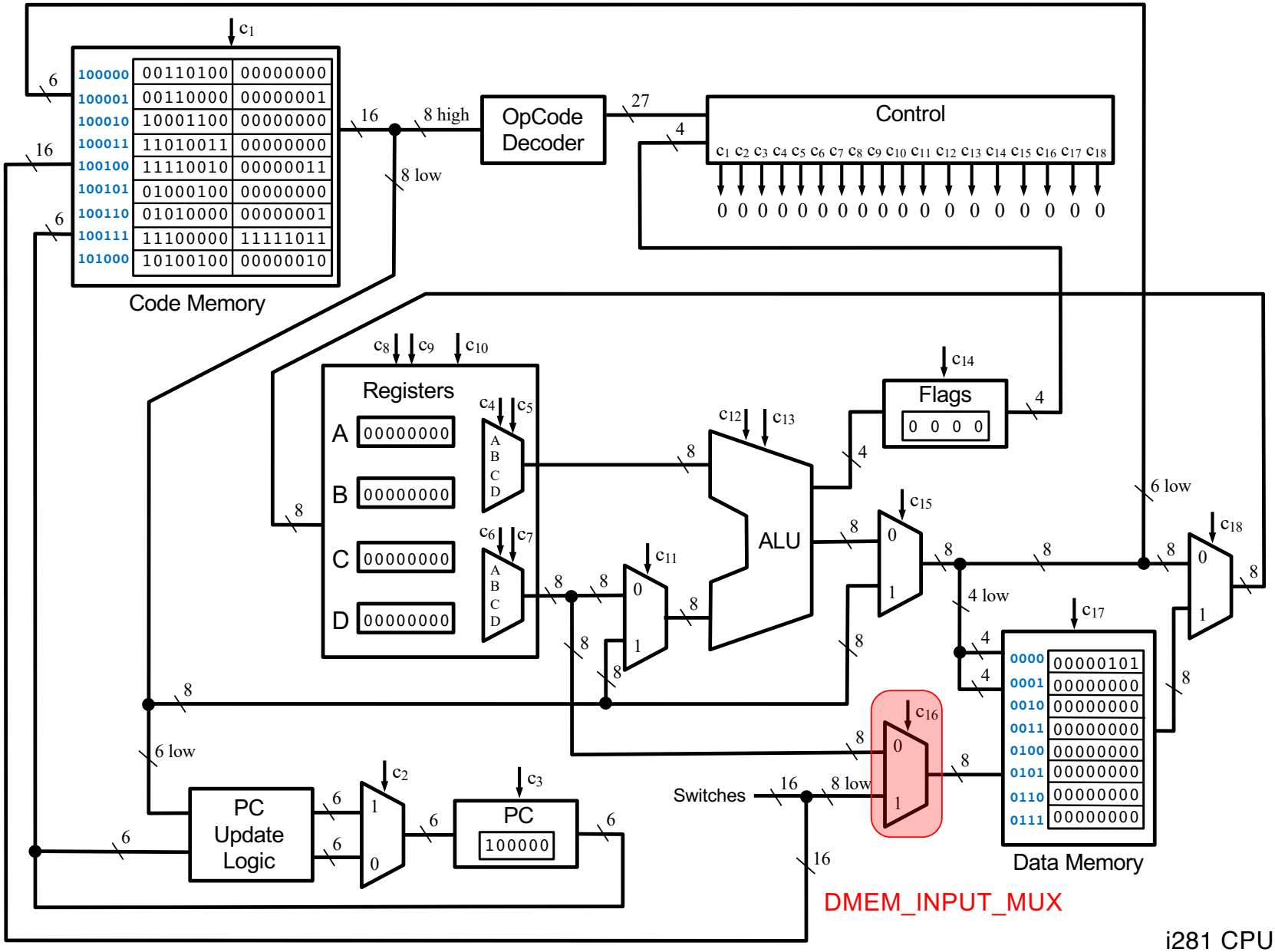

C11
U0
V0
0
1
Z0
U1
V1
Z1
U2
V2
Z2
U3
V3
Z3
U4
V4
Z4
U5
V5
Z5
U6
V6
Z6
U7
V7
Z7

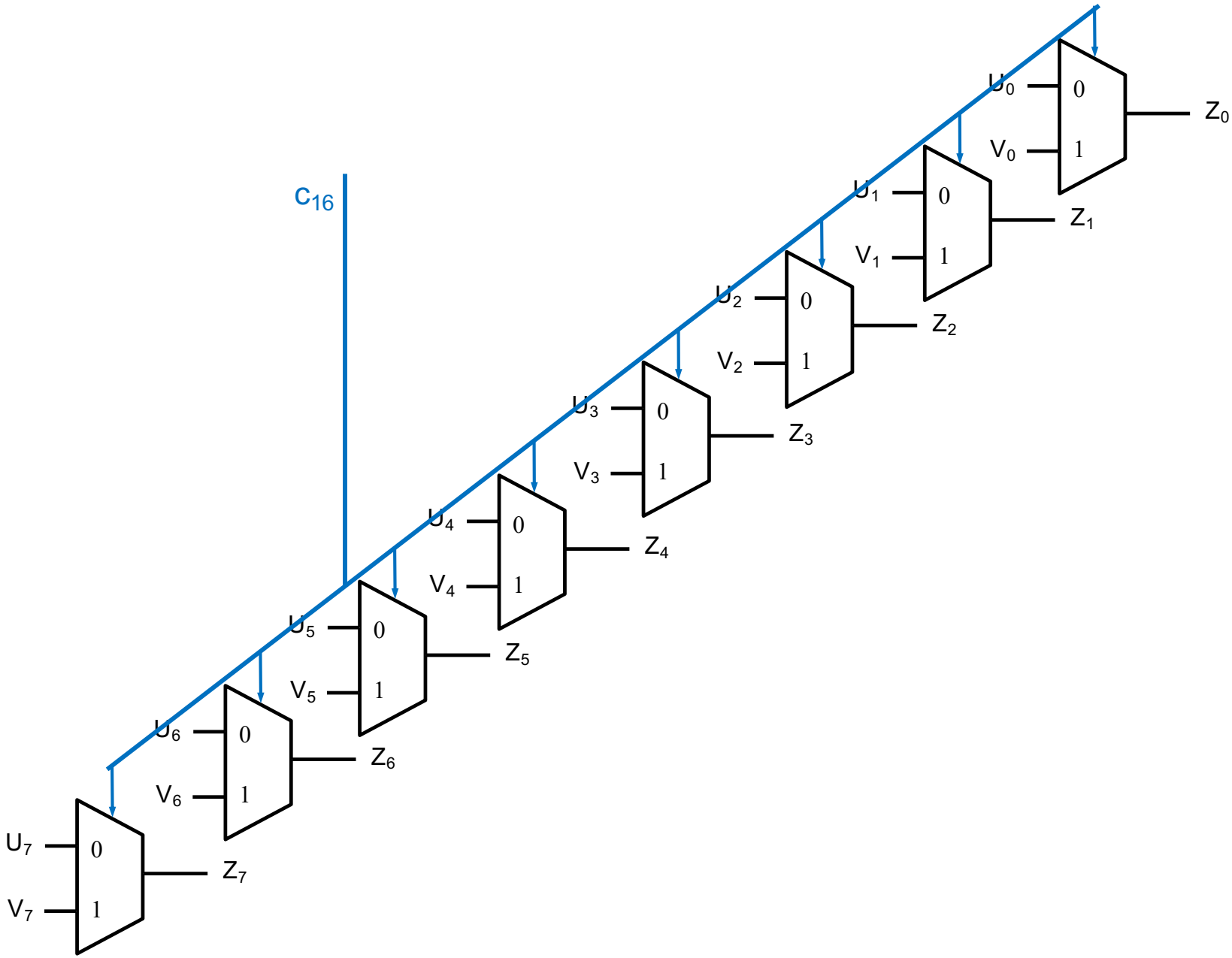

0
c11
U0
V0
Z0
U1
V1
Z1
U2
V2
Z2
U3
V3
Z3
U4
V4
Z4
U5
V5
Z5
U6
V6
Z6
U7
V7
Z7
0
1

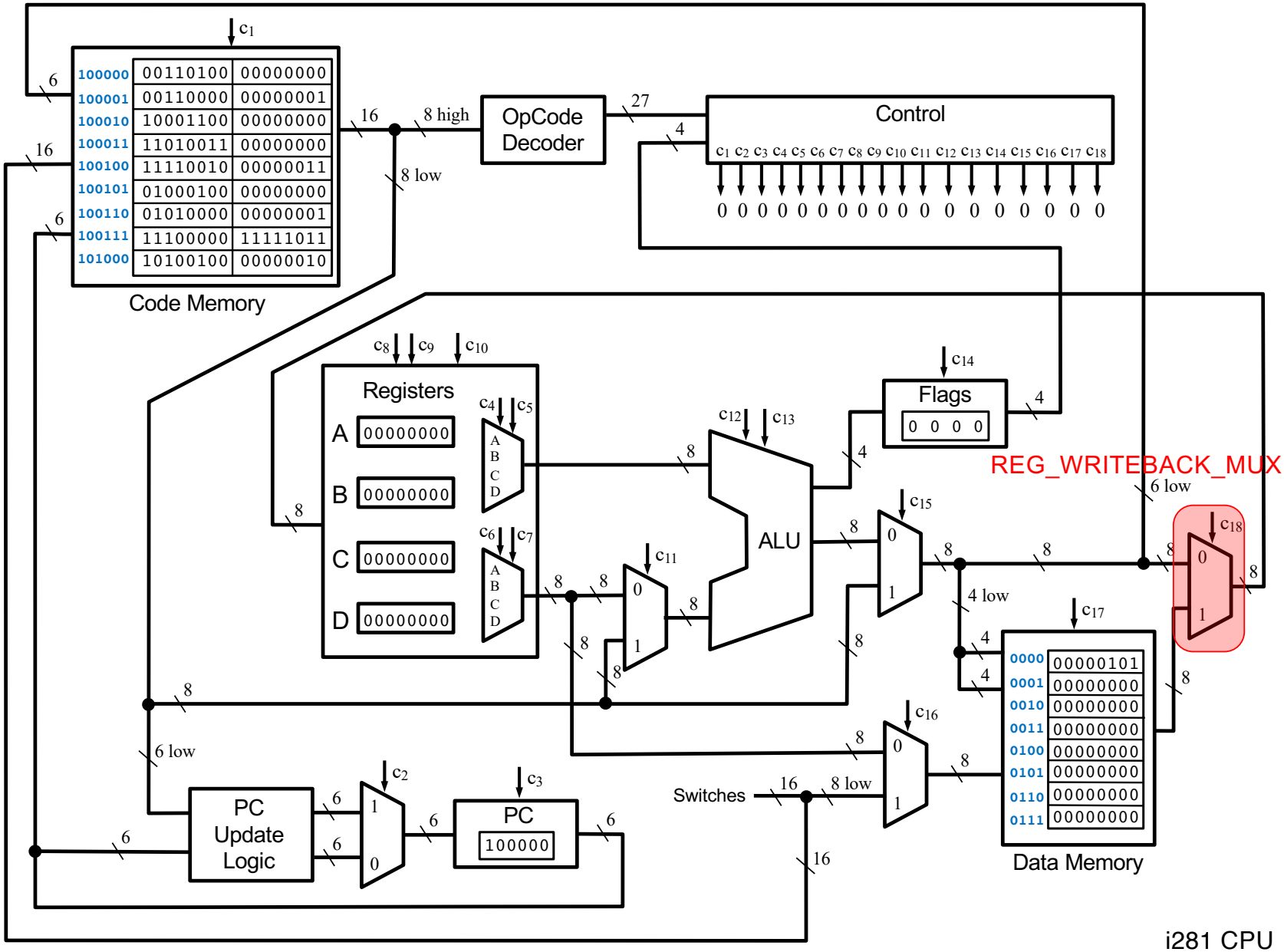

1
C11
U0
V0
Z0
U1
V1
Z1
U2
V2
Z2
U3
V3
Z3
U4
V4
Z4
U5
V5
Z5
U6
V6
Z6
U7
V7
Z7
0
1

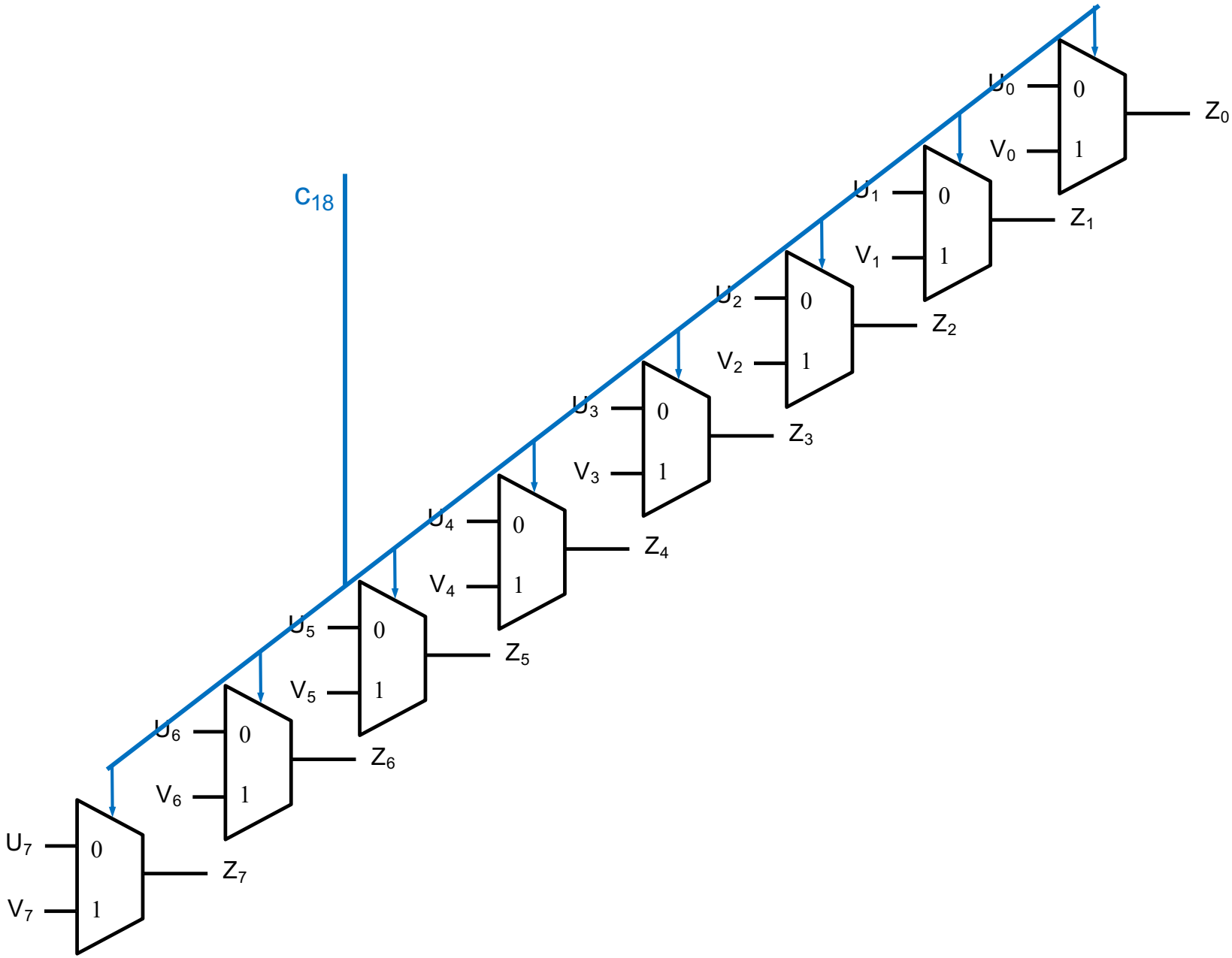

c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c12 c13
ALU
c14
Flags
0 0 0 0
4
ALU_RESULT_MUX
c15
c11
c16
c17
c18
6 low
4 low
8 low
Switches
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
PC Update Logic
c2
c3
PC
100000
i281 CPU

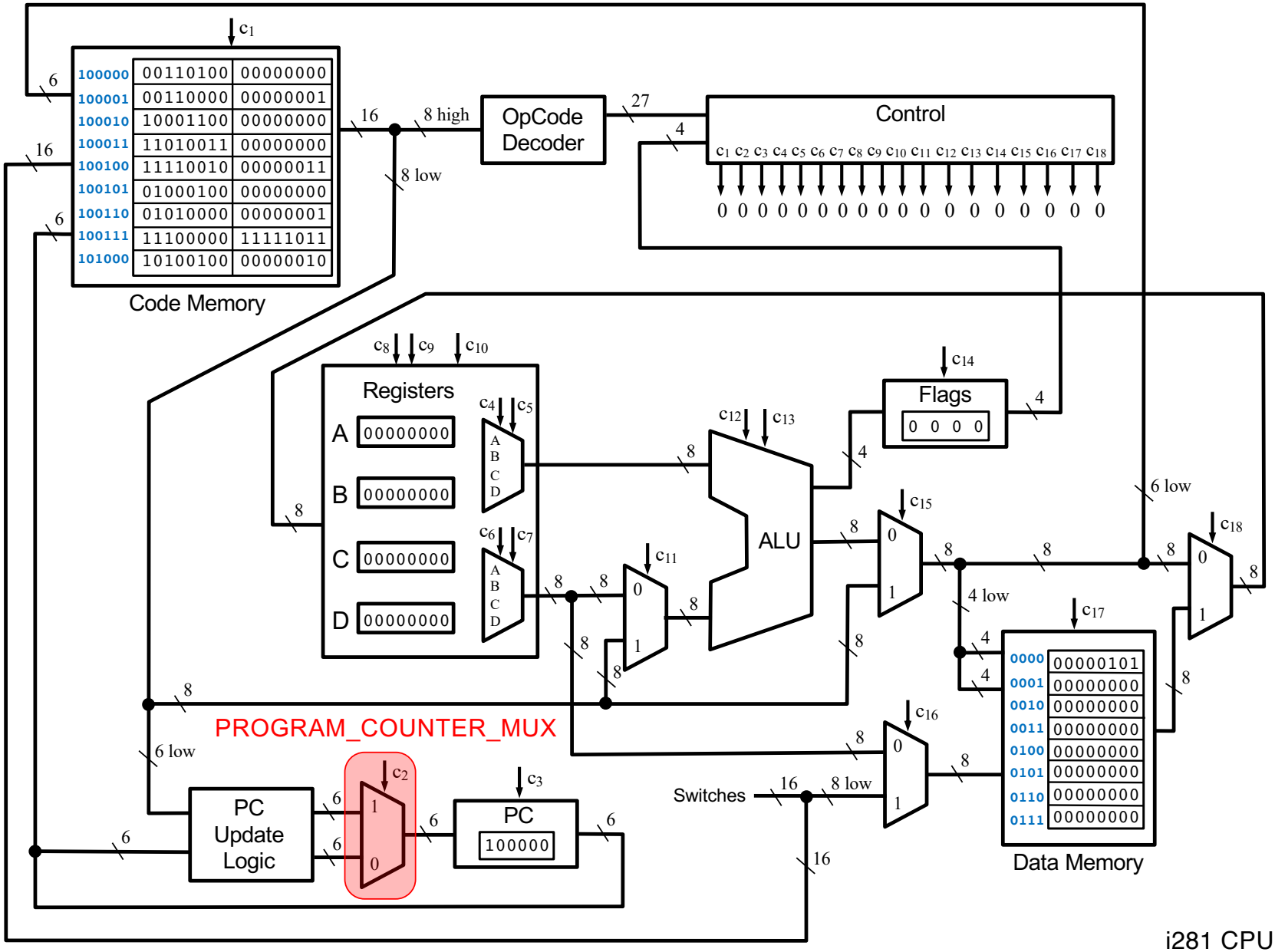

C15
U0
V0
0
1
Z0
U1
V1
Z1
U2
V2
Z2
U3
V3
Z3
U4
V4
Z4
U5
V5
Z5
U6
V6
Z6
U7
V7
Z7

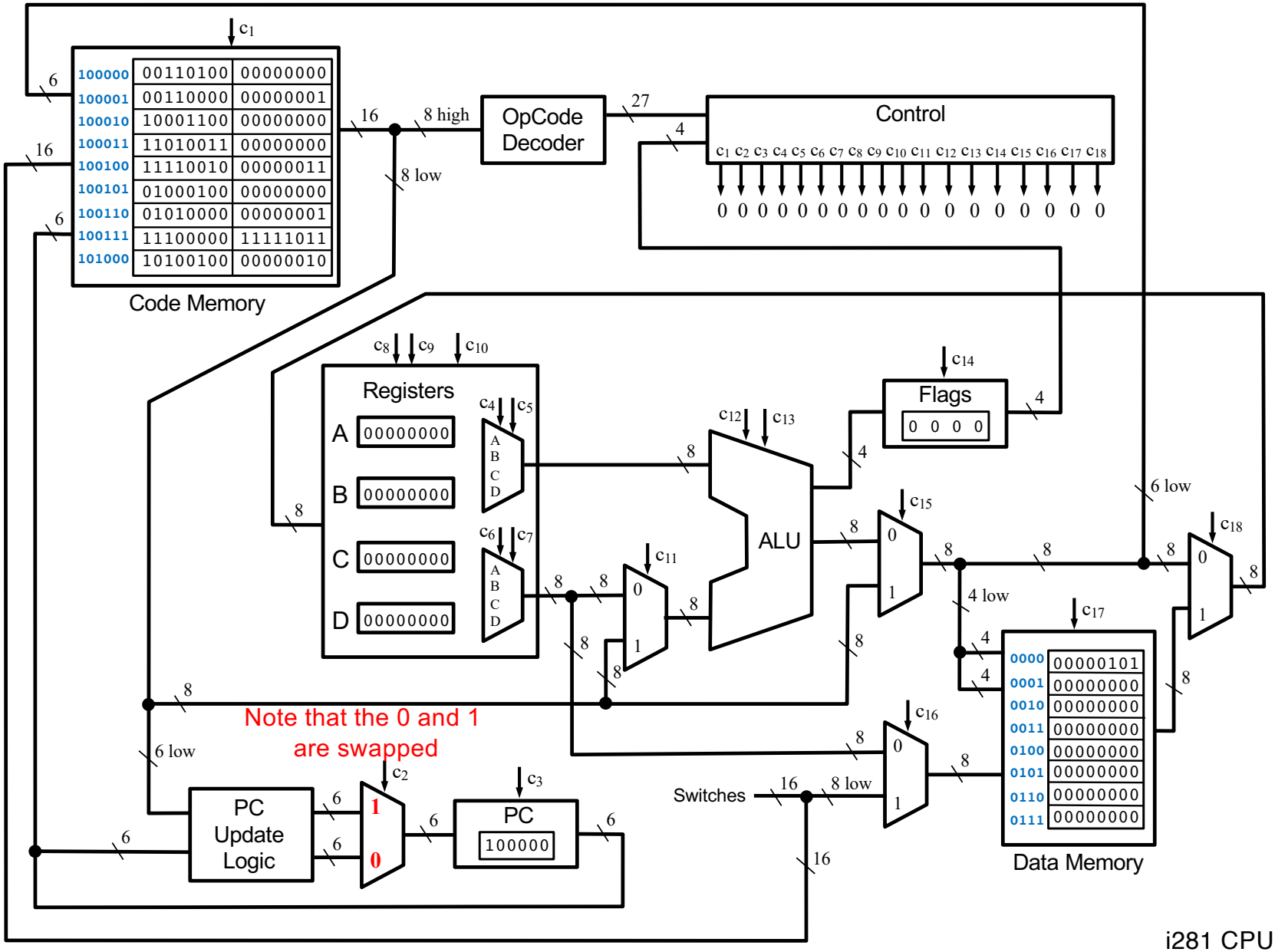

c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c14
Flags
0 0 0 0
4
c12 c13
ALU
c15
c11
c16
c17
c18
6 low
4 low
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
Switches
16
8 low
DMEM_INPUT_MUX
PC Update Logic
c2
c3
PC
100000
6 low
i281 CPU


C16
U0
V0
0
1
Z0
U1
V1
Z1
U2
V2
Z2
U3
V3
Z3
U4
V4
Z4
U5
V5
Z5
U6
V6
Z6
U7
V7
Z7

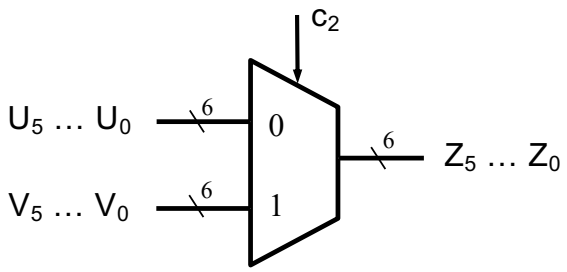

c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c14
Flags
0 0 0 0
4
c12 c13
ALU
c15
c11
c16
c17
c18
REG_WRITEBACK_MUX
6 low
4 low
8 low
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
Switches
16
PC Update Logic
c2
c3
PC
100000
6
6 low
i281 CPU


C18
U0
V0
0
1
Z0
U1
V1
Z1
U2
V2
Z2
U3
V3
Z3
U4
V4
Z4
U5
V5
Z5
U6
V6
Z6
U7
V7
Z7

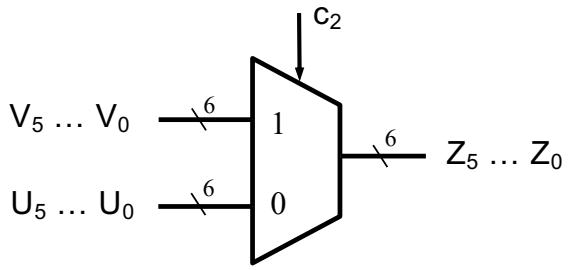

c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c12 c13
ALU
c14
Flags
0 0 0 0
4
c15
c11
c16
c17
c18
6 low
4 low
8 low
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
PROGRAM_COUNTER_MUX
c2
c3
PC Update Logic
PC
100000
Switches
16
i281 CPU


c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c12 c13
ALU
c14
Flags
0 0 0 0
4
c15
c11
c16
c17
c18
6 low
4 low
8 low
Switches
16
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
Note that the 0 and 1
are swapped
c2
c3
PC
Update
Logic
PC
100000
1
0
i281 CPU

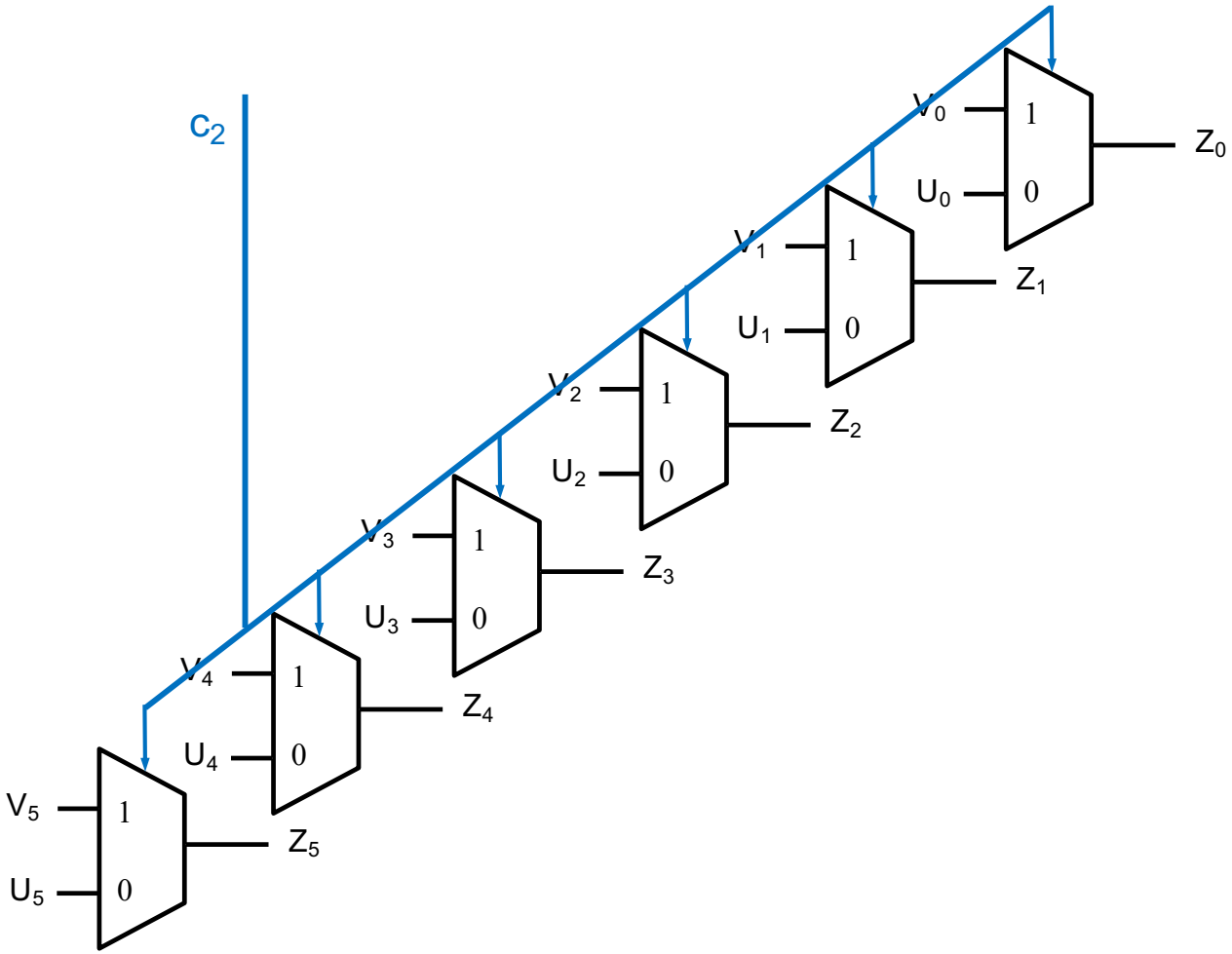
2-to-1 Bus Multiplexer (with 6-bit lines)

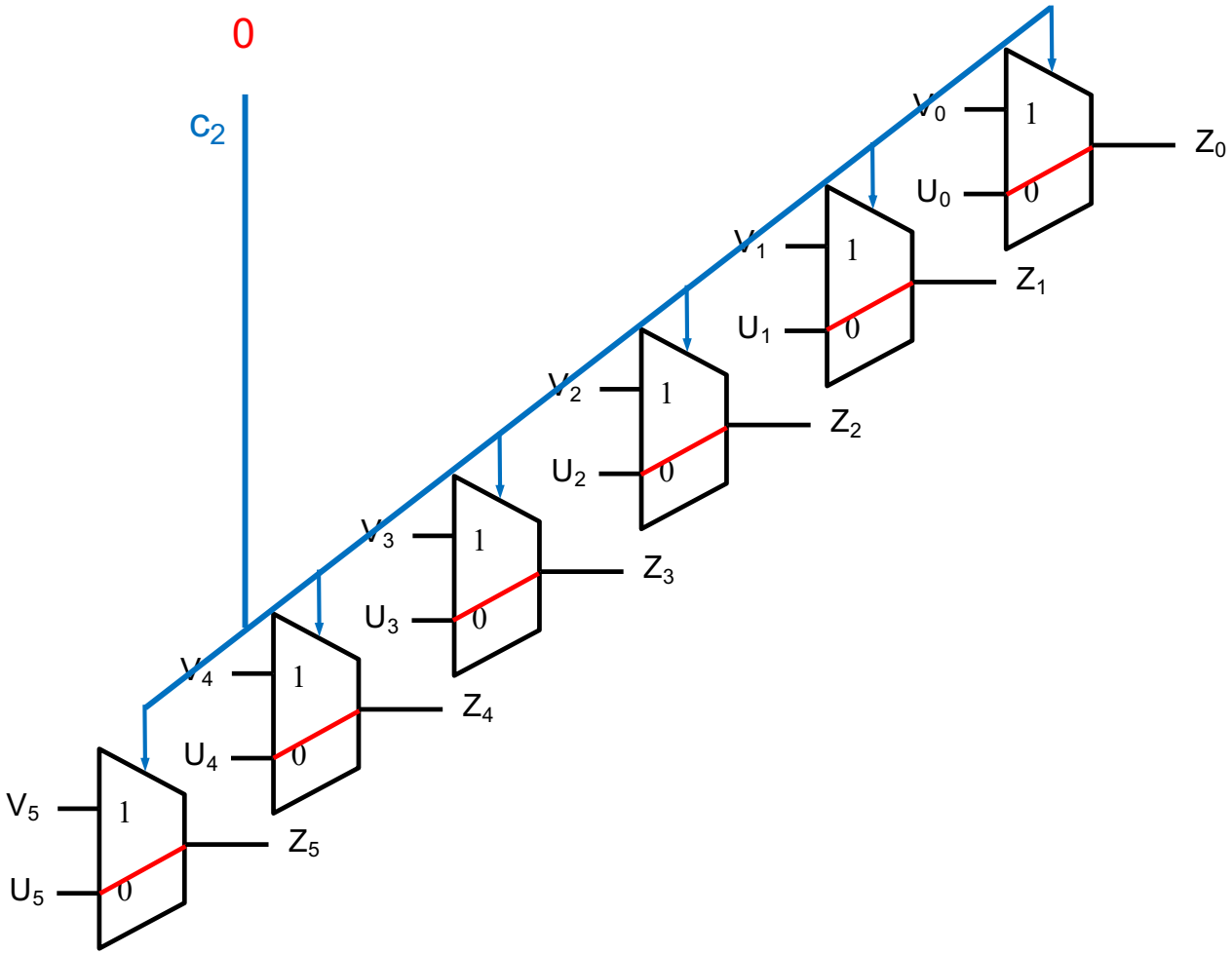


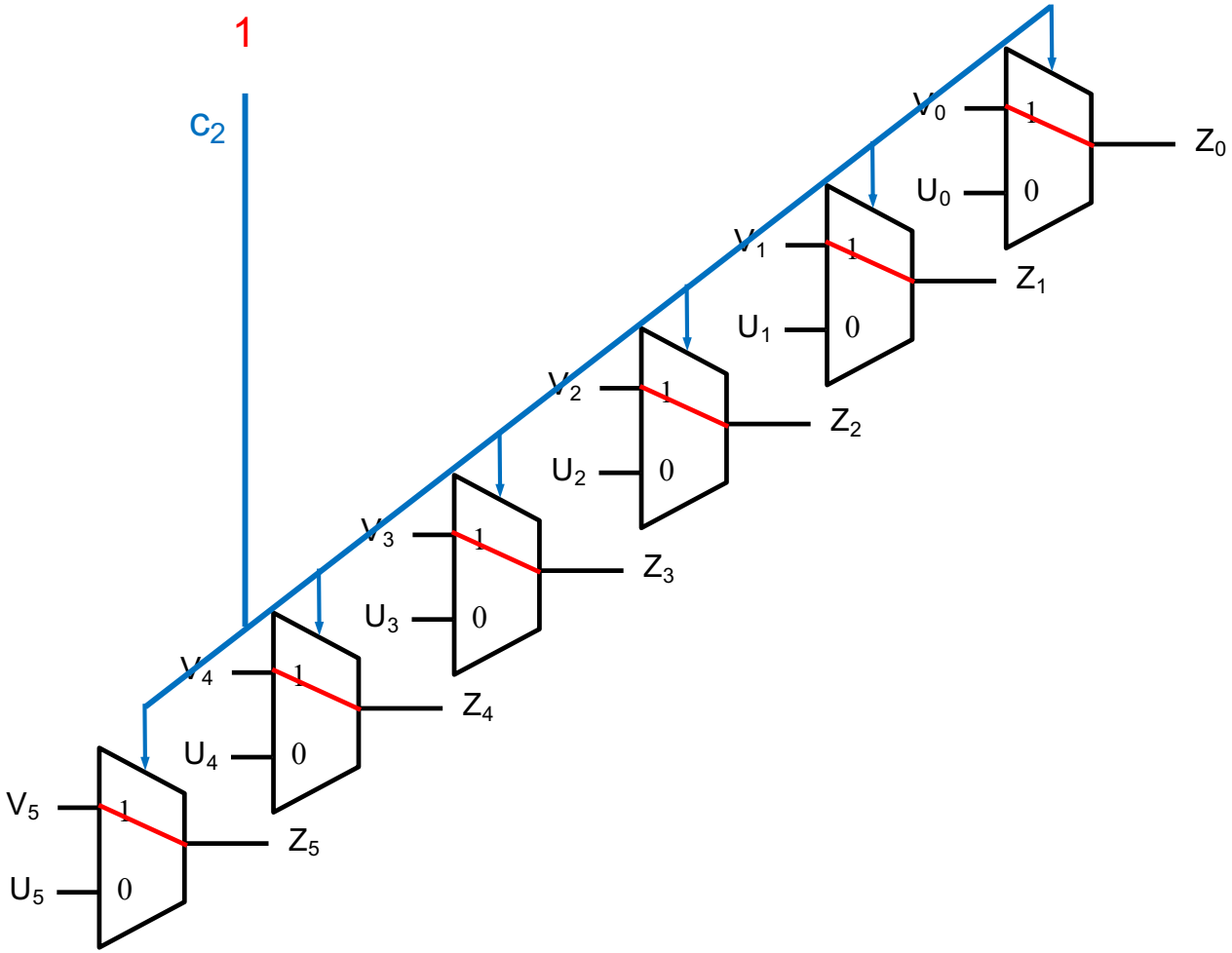
2-to-1 Bus Multiplexer (with 6-bit lines)



Note that the 0 and the 1 are swapped
(in order to simplify the large diagram).

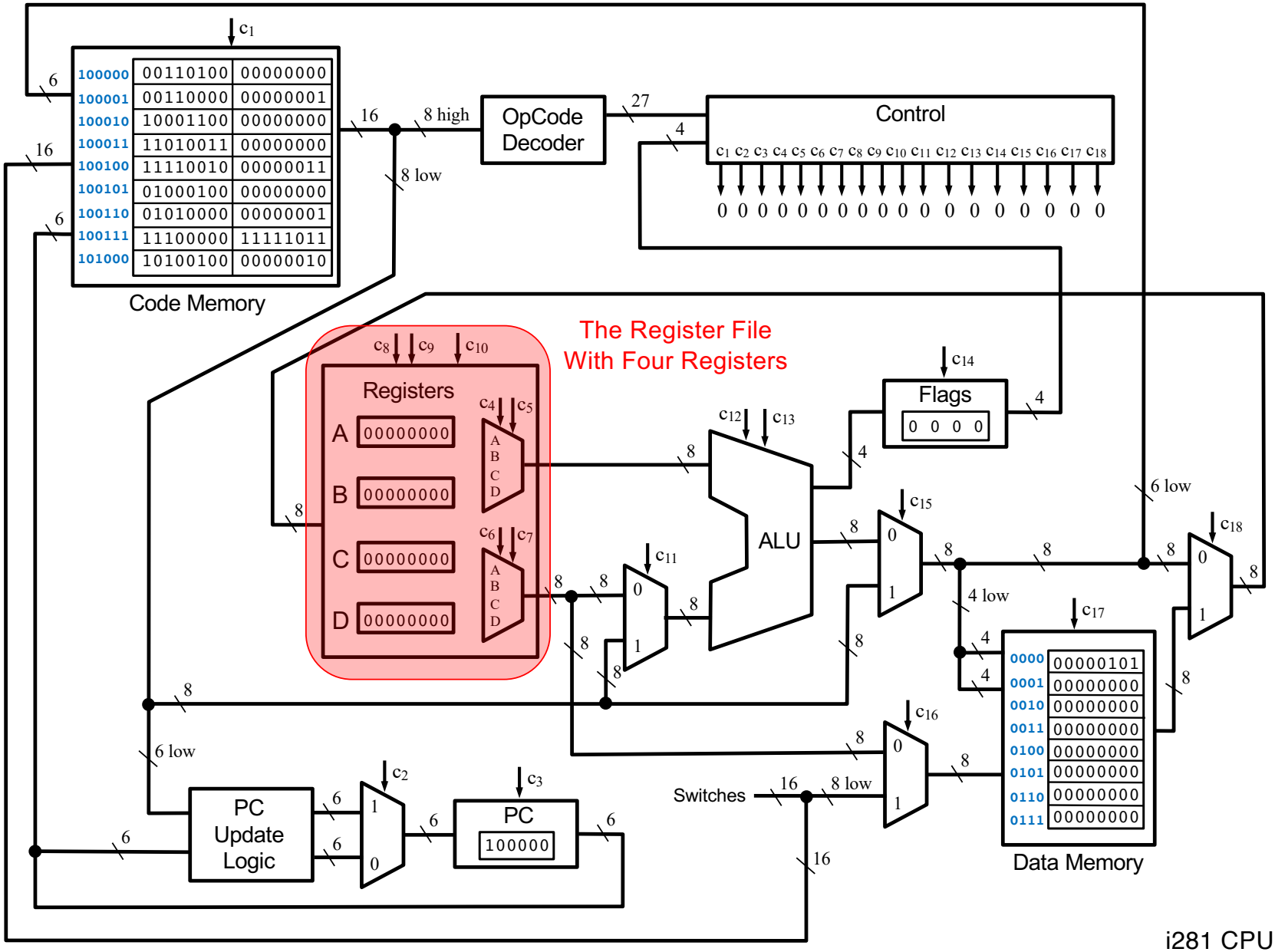

C2
V0
U0
1
0
Z0
V1
U1
Z1
V2
U2
Z2
V3
U3
Z3
V4
U4
Z4
V5
U5
Z5

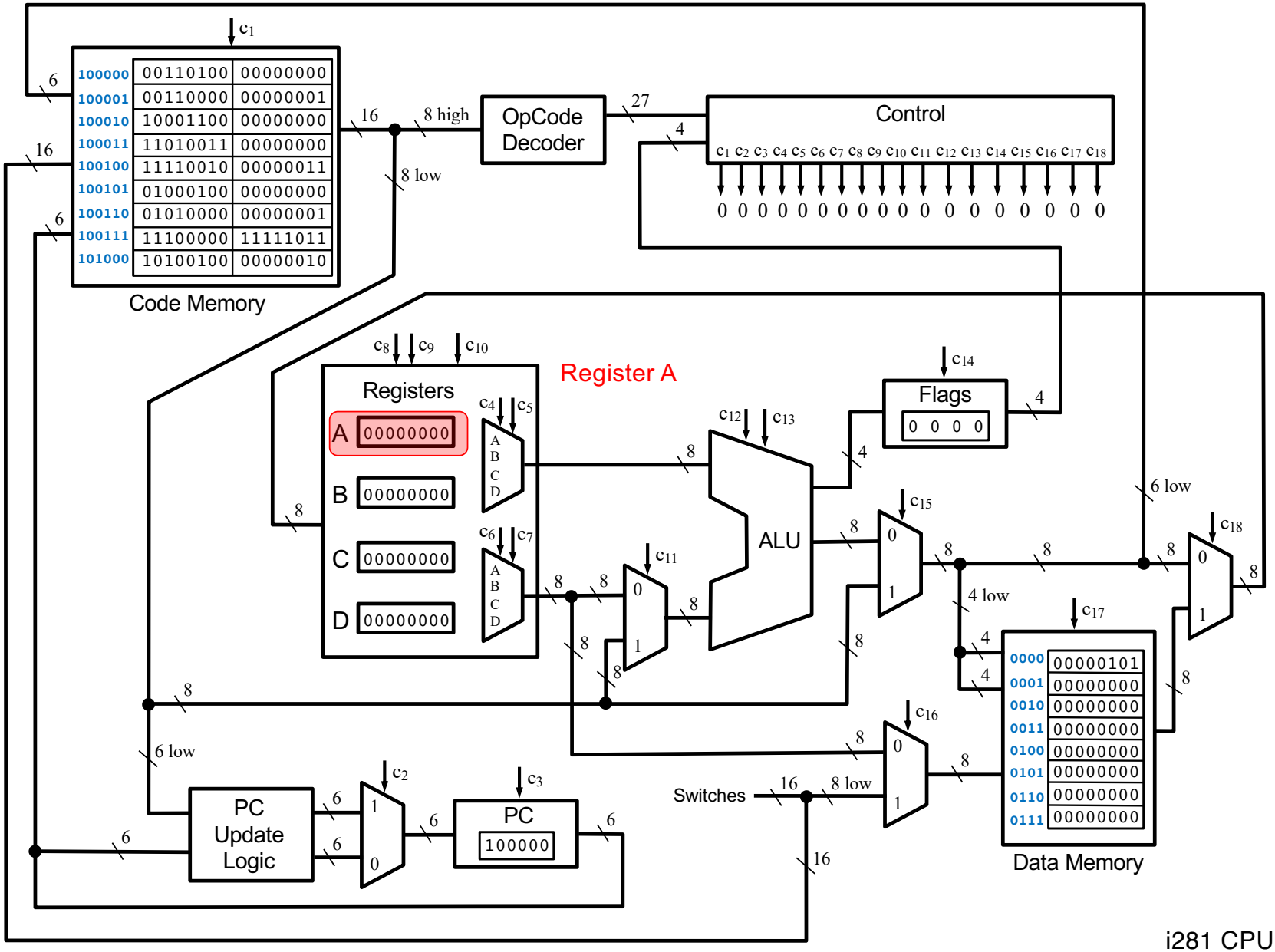

0
c2
V0
U0
1
0
Z0
V1
U1
Z1
V2
U2
Z2
V3
U3
Z3
V4
U4
Z4
V5
U5
Z5

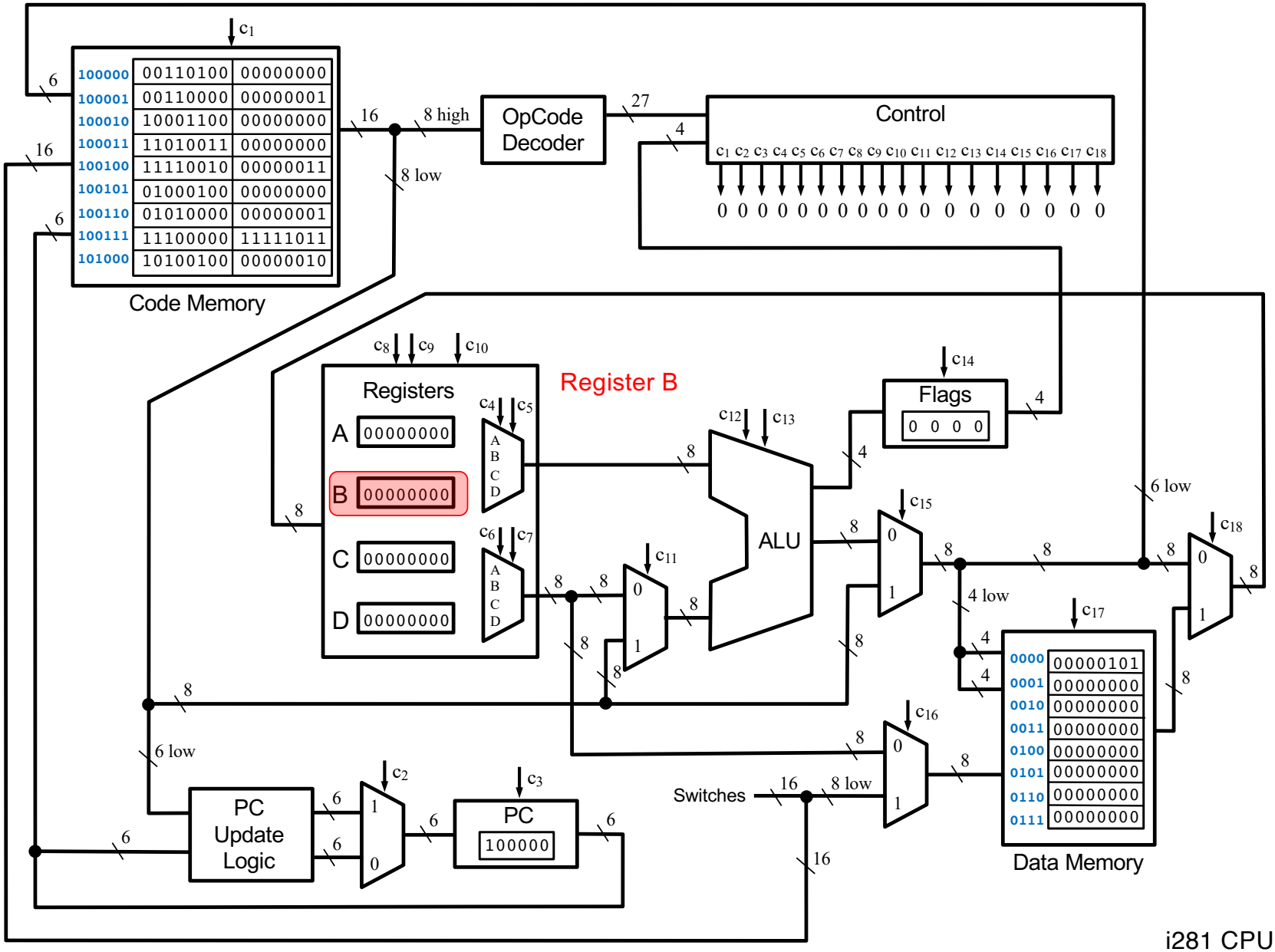

1
c2
V0
1
Z0
U0
0
V1
1
Z1
U1
0
V2
1
Z2
U2
0
V3
1
Z3
U3
0
V4
1
Z4
U4
0
V5
1
Z5
U5
0

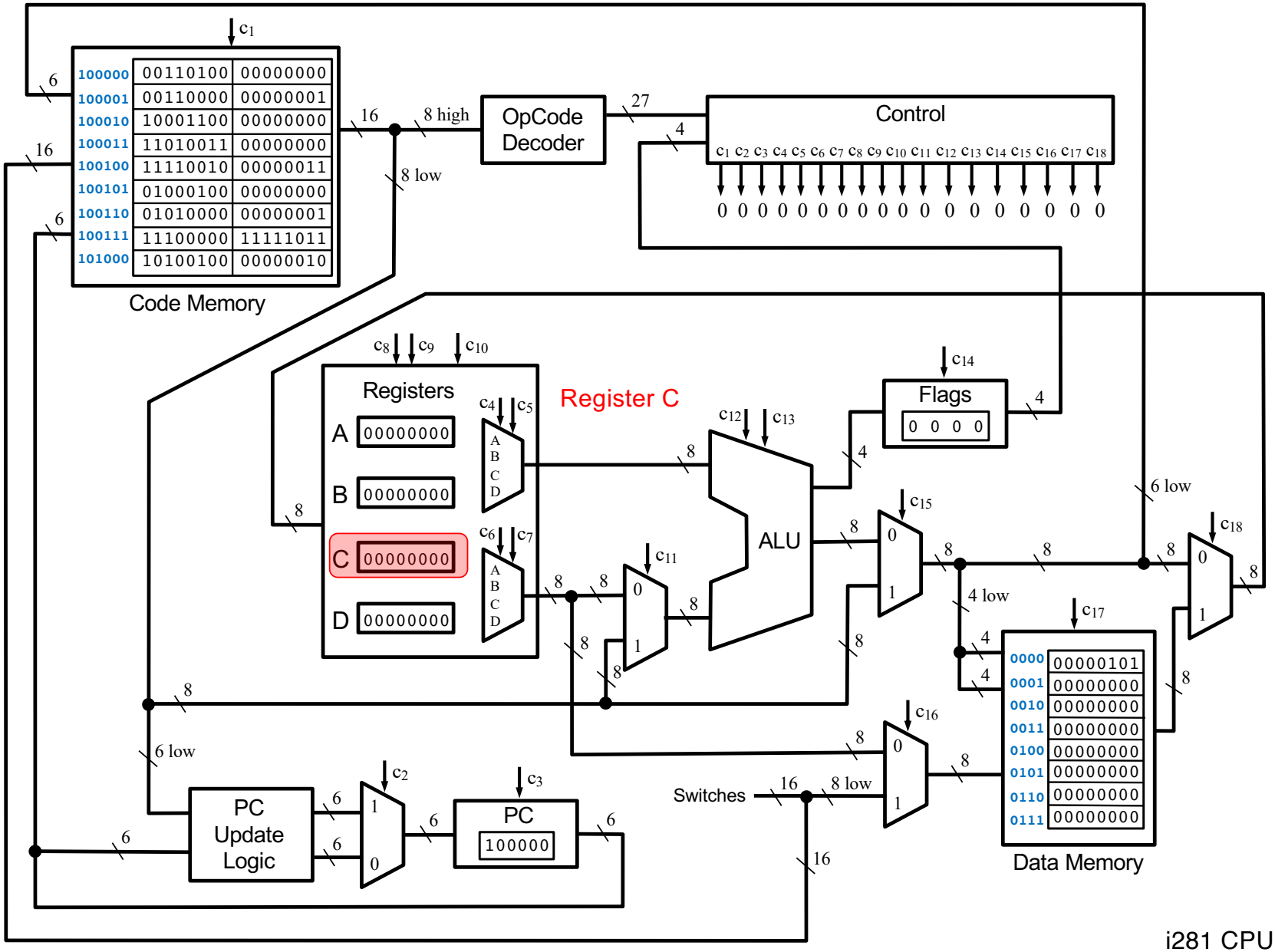
The Four Registers

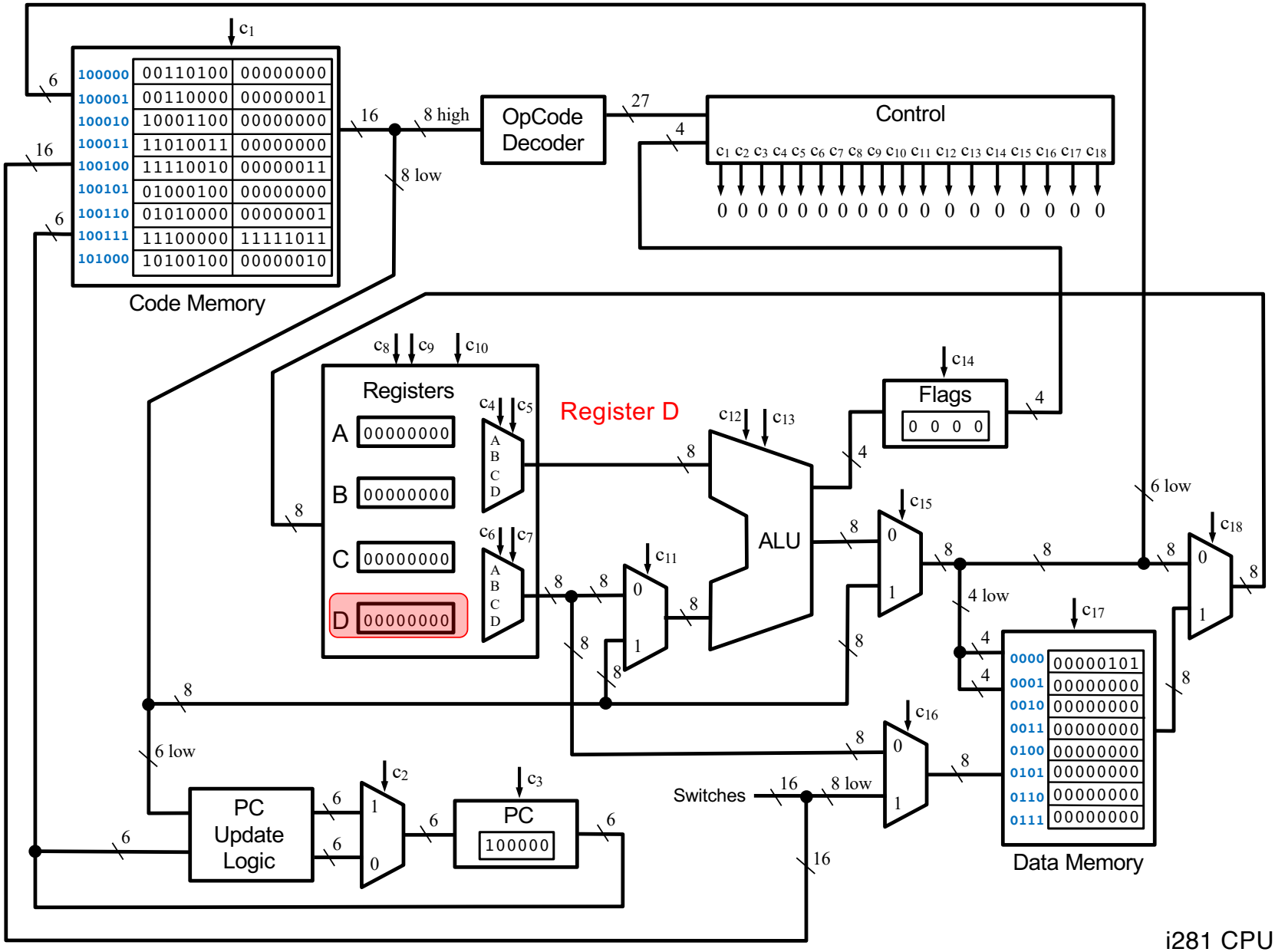
(implemented as a register file with two read ports and one write port)


c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
The Register File
With Four Registers
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
c14
Flags
0 0 0 0
c12 c13
ALU
c15
c11
c16
c17
c18
6 low
4 low
8 low
Switches
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
PC Update Logic
c2
c3
PC
100000
i281 CPU

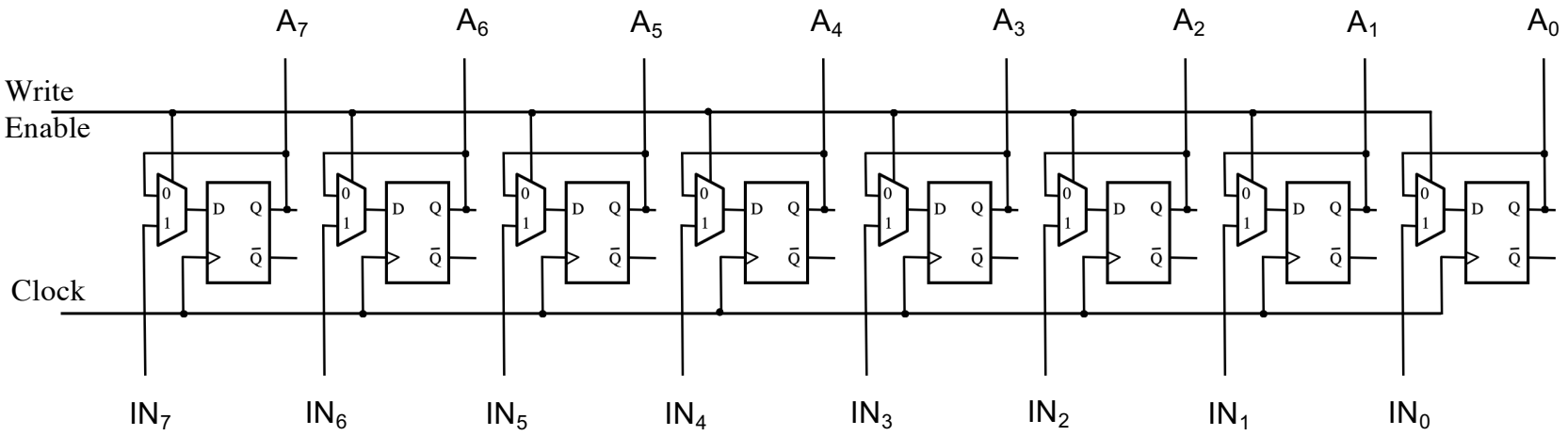

c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Register A
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
c11
c12 c13
ALU
c14
Flags
0 0 0 0
c15
c16
c17
c18
6 low
4 low
8 low
Switches
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
c2
c3
PC Update Logic
PC
100000
i281 CPU


c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
Register B
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c12 c13
ALU
c14
Flags
0 0 0 0
4
c15
c11
c16
c17
c18
6 low
4 low
8 low
Switches
16
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
c2
c3
PC Update Logic
PC
100000
i281 CPU


c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
Register C
c12 c13
ALU
c14
Flags
0 0 0 0
4
c15
c11
c16
c17
c18
6 low
4 low
8 high
Switches
16
8 low
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
c2
c3
PC Update Logic
PC
100000
i281 CPU

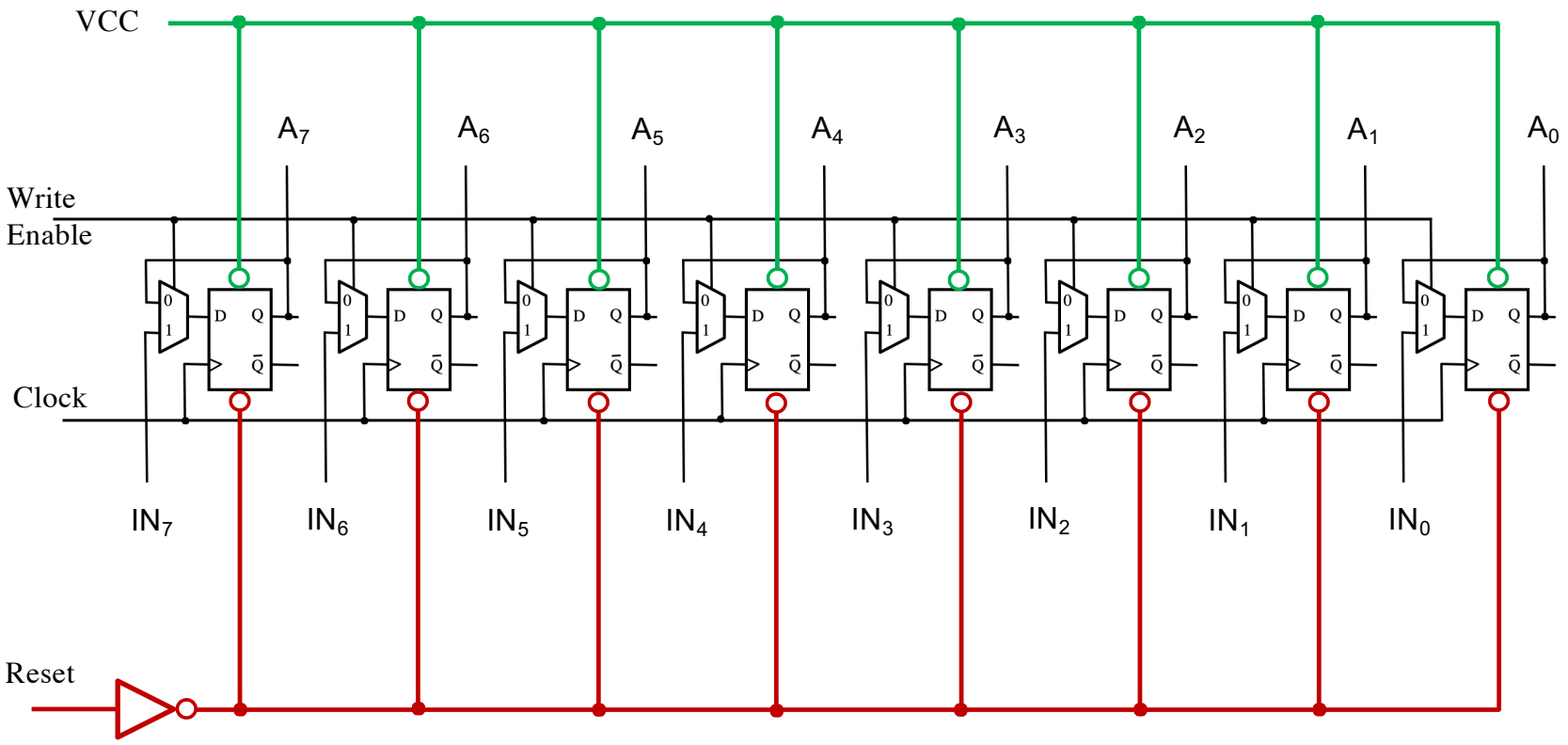

c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
Register D
c12 c13
ALU
c14
Flags
0 0 0 0
4
c15
c11
c16
c17
c18
6 low
4 low
8 high
Switches
16
8 low
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
c2
c3
PC Update Logic
PC
100000
6 low
i281 CPU

Register A

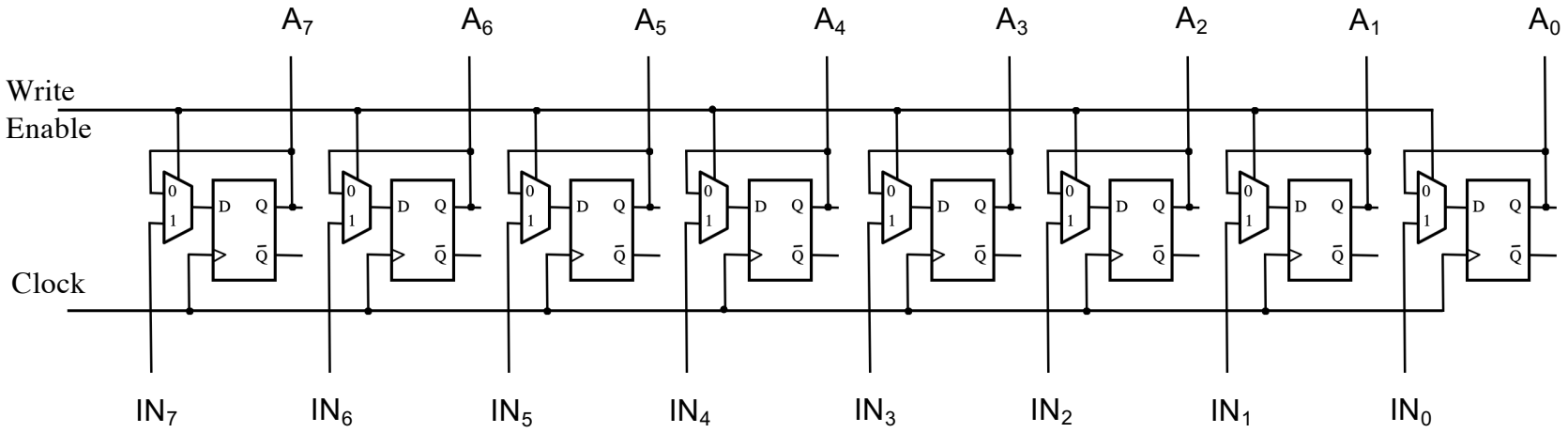


8-Bit Parallel-Access Register

Register A

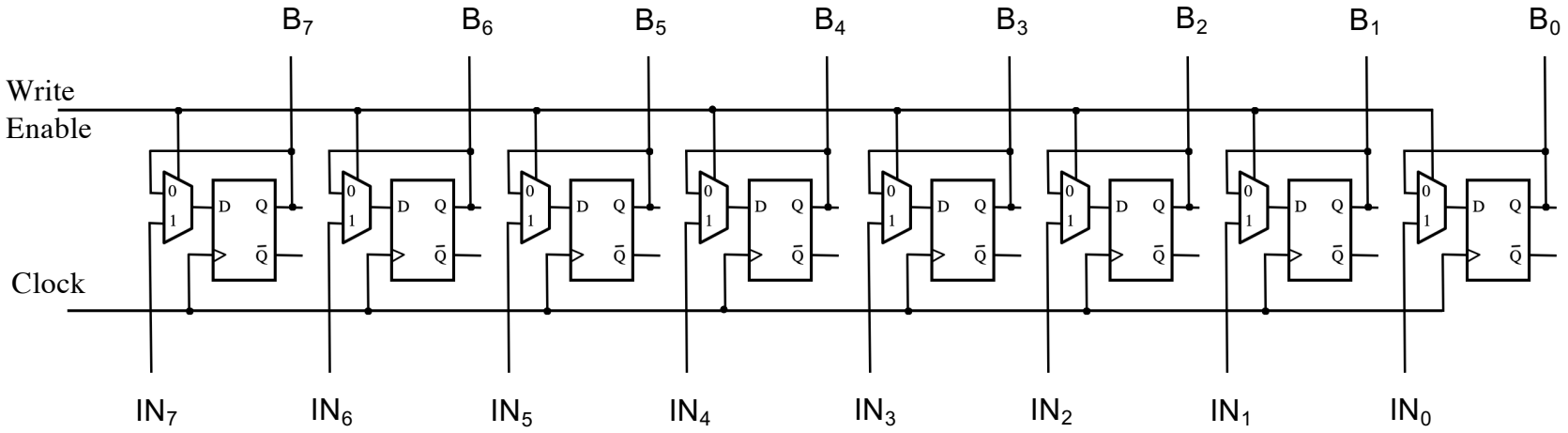

VCC
A7
A6
A5
A4
A3
A2
A1
A0
Write Enable
Clock
IN7
IN6
IN5
IN4
IN3
IN2
IN1
IN0
Reset
D
Q
Q̄
0
1

Register A



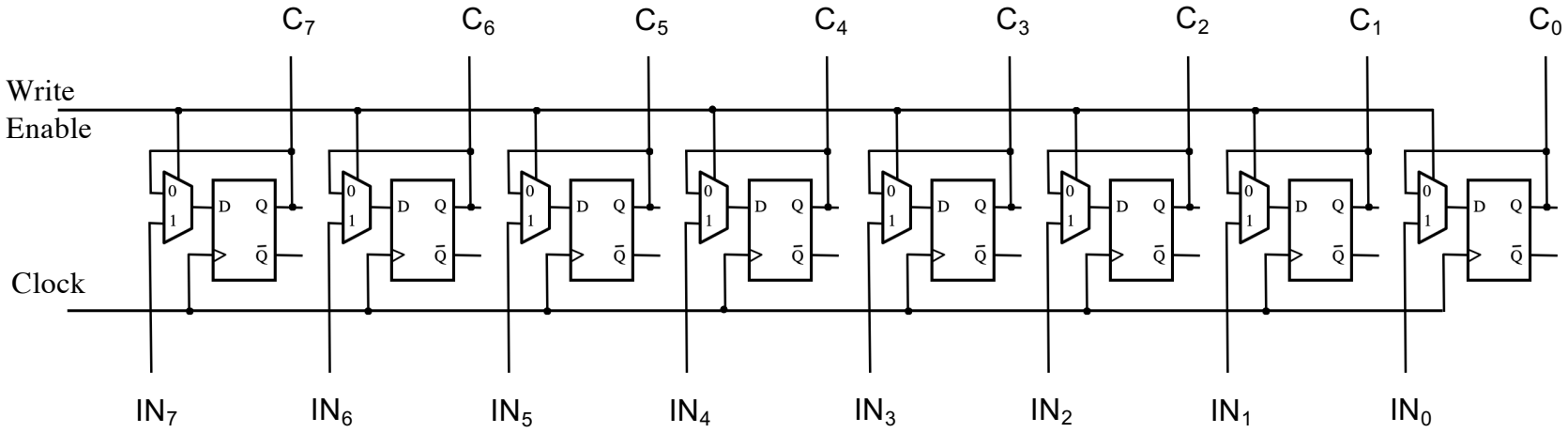
8-Bit Parallel-Access Register

Register B



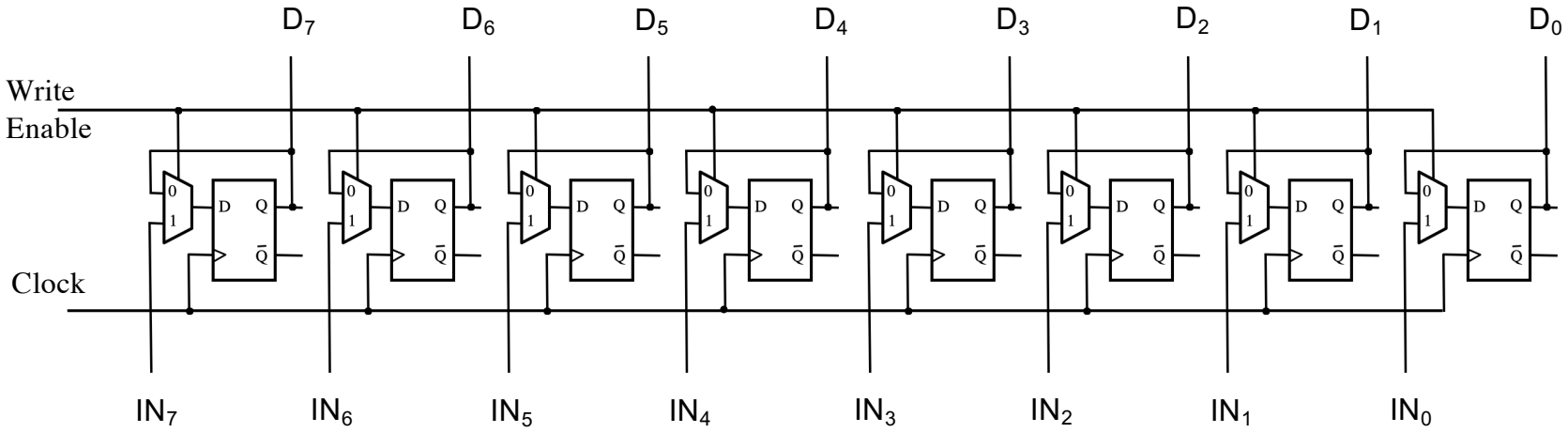
8-Bit Parallel-Access Register

Register C

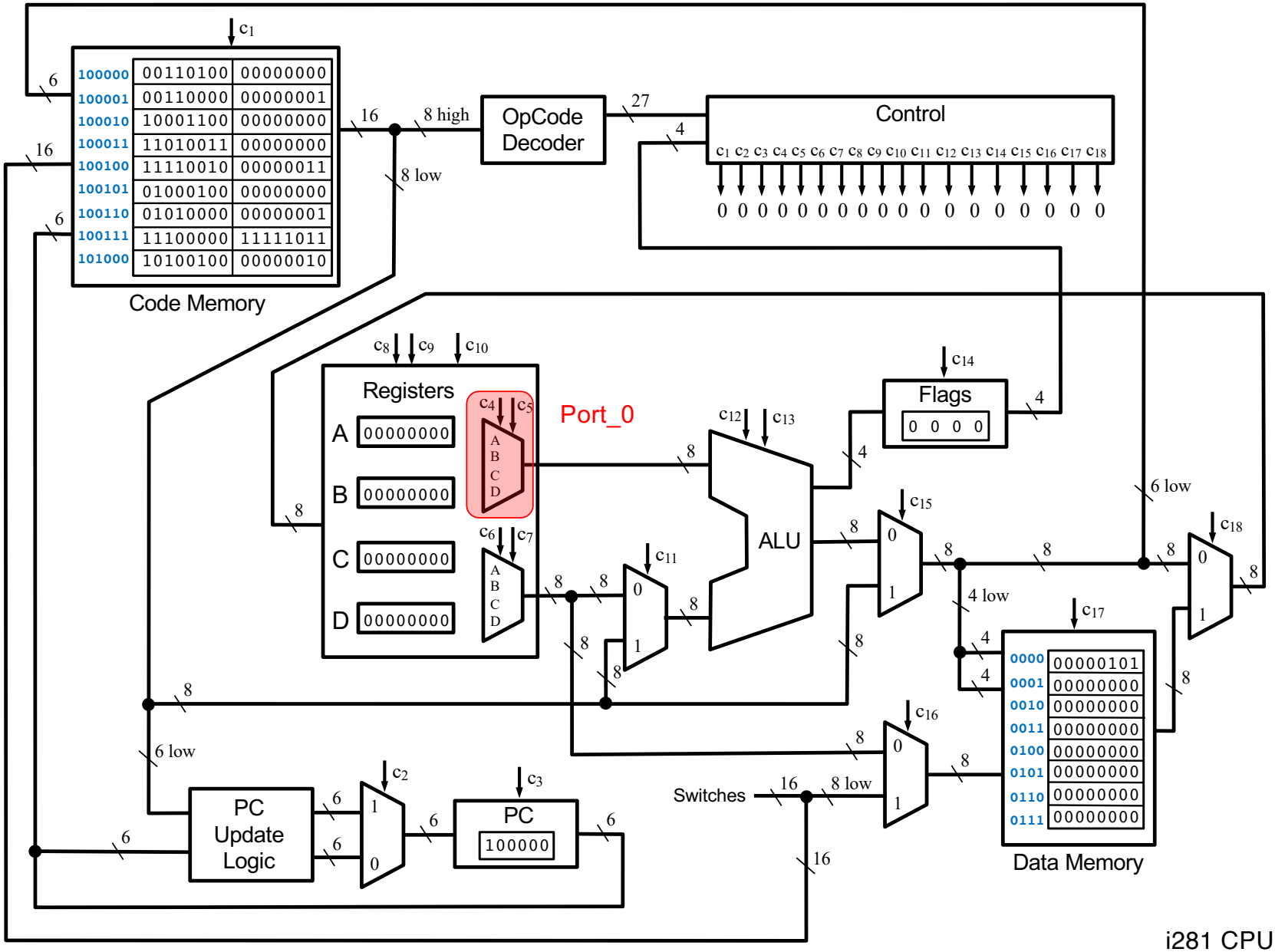


8-Bit Parallel-Access Register

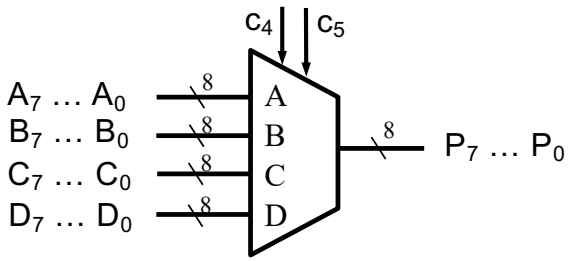
Register D

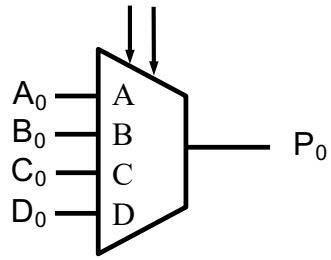


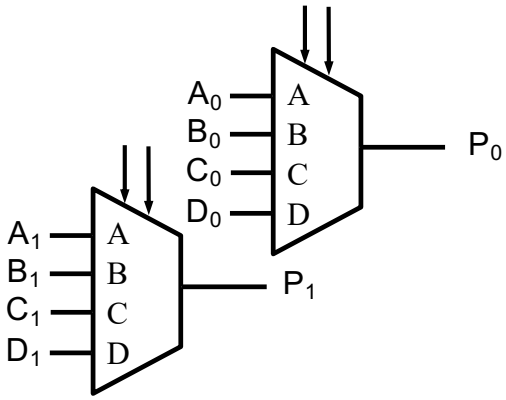
8-Bit Parallel-Access Register

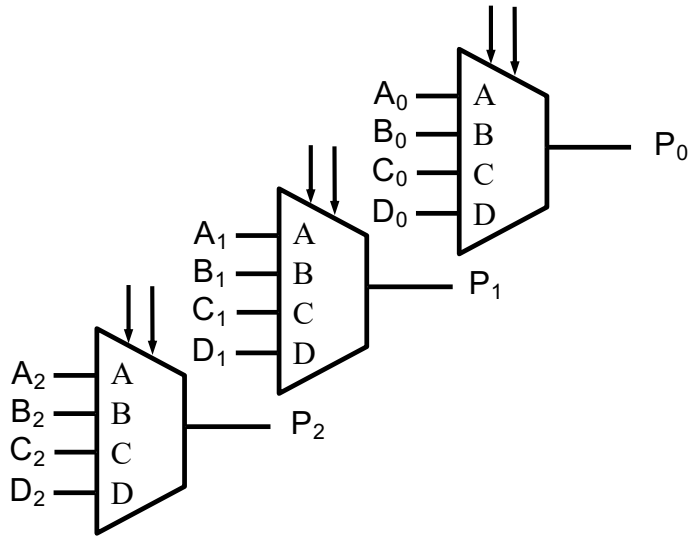

c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
Port_0
8
c12 c13
ALU
c14
Flags
0 0 0 0
4
c15
c11
c16
c17
c18
6 low
4 low
8 low
Switches
16
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
PC Update Logic
c2
c3
PC
100000
i281 CPU

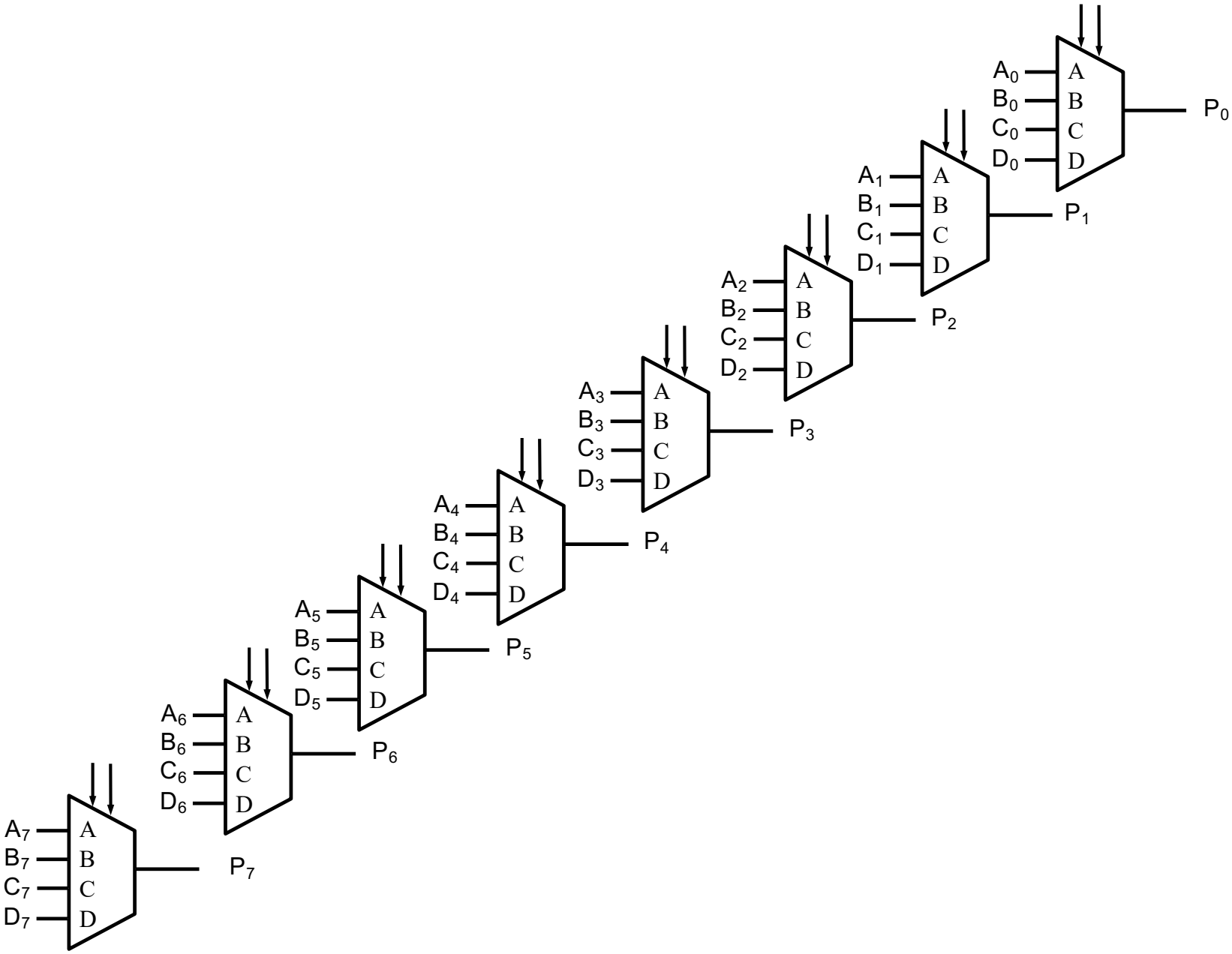
4-to-1 Bus Multiplexer (with 8-bit lines)

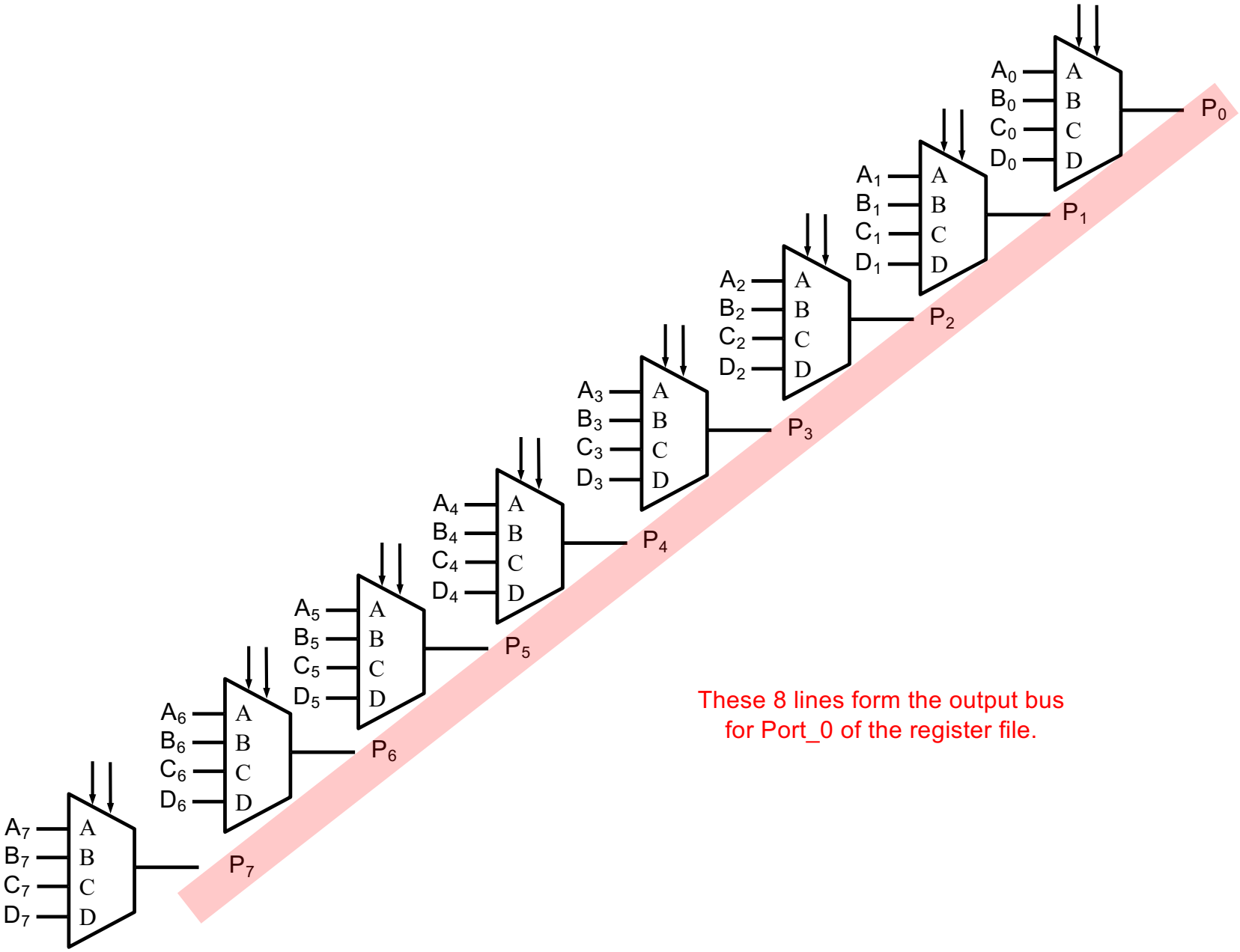


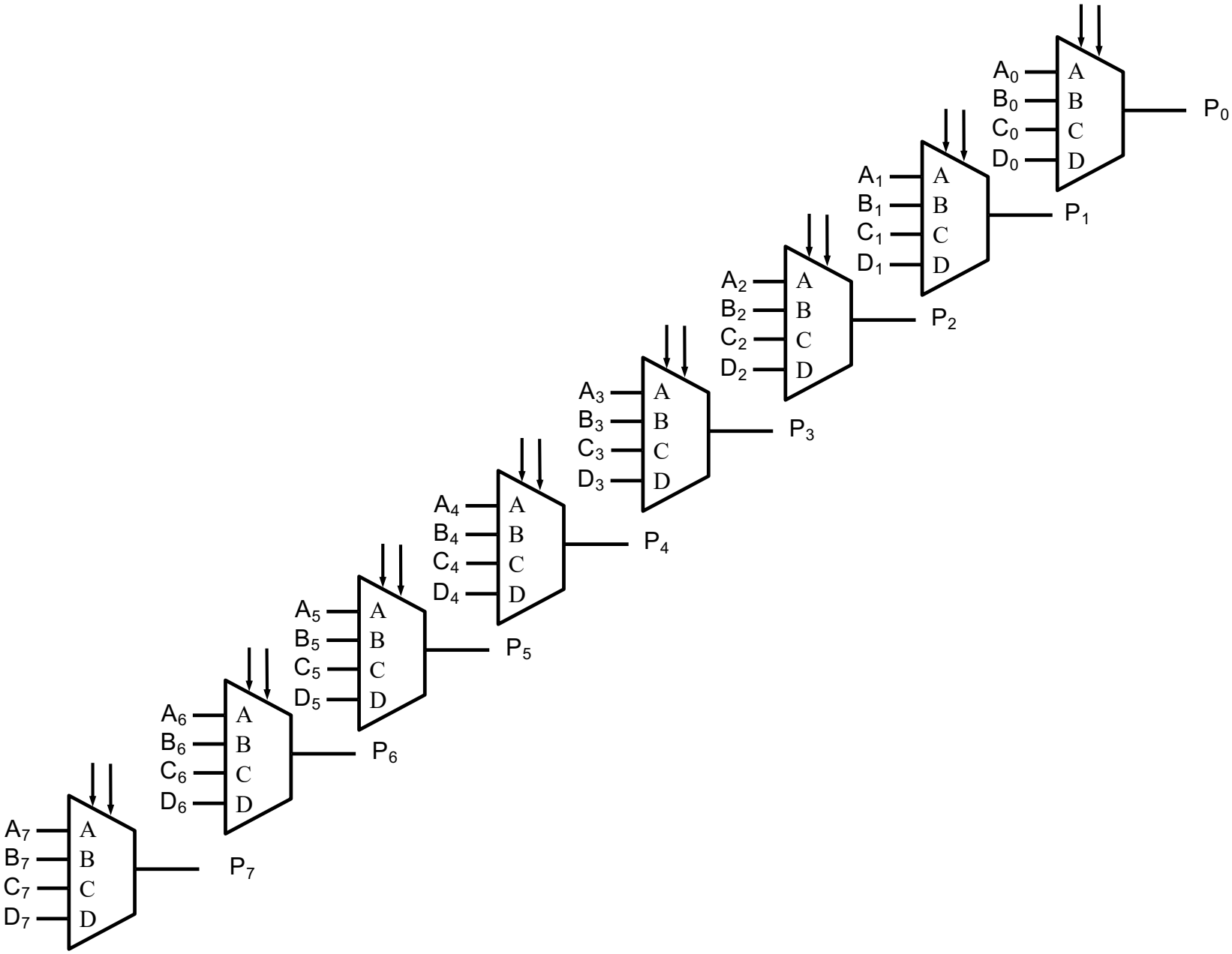

A0
B0
C0
D0
A
B
C
D
P0

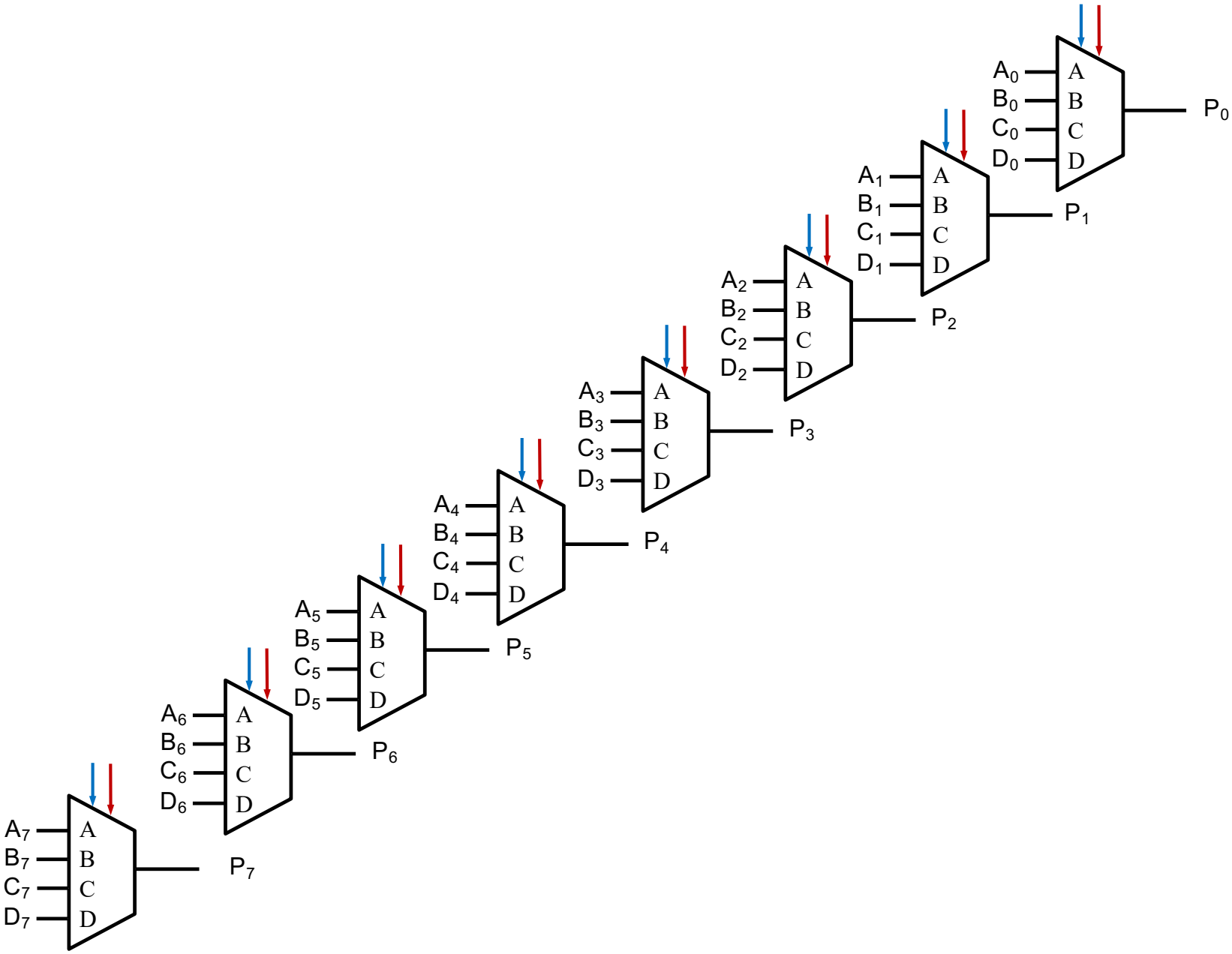

A0
B0
C0
D0
A
B
C
D
P0
A1
B1
C1
D1
A
B
C
D
P1

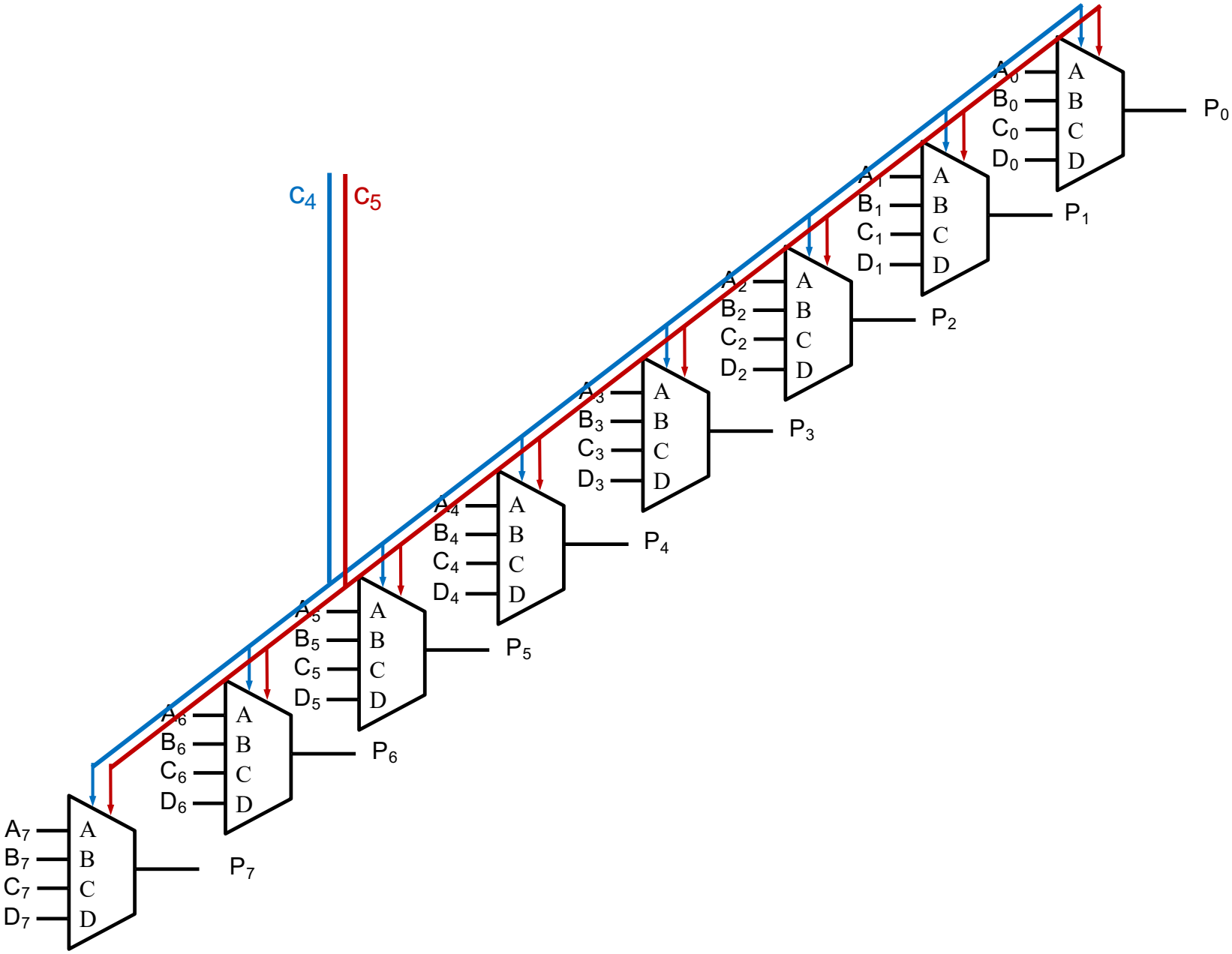

A0
B0
C0
D0
A
B
C
D
P0
A1
B1
C1
D1
A
B
C
D
P1
A2
B2
C2
D2
A
B
C
D
P2

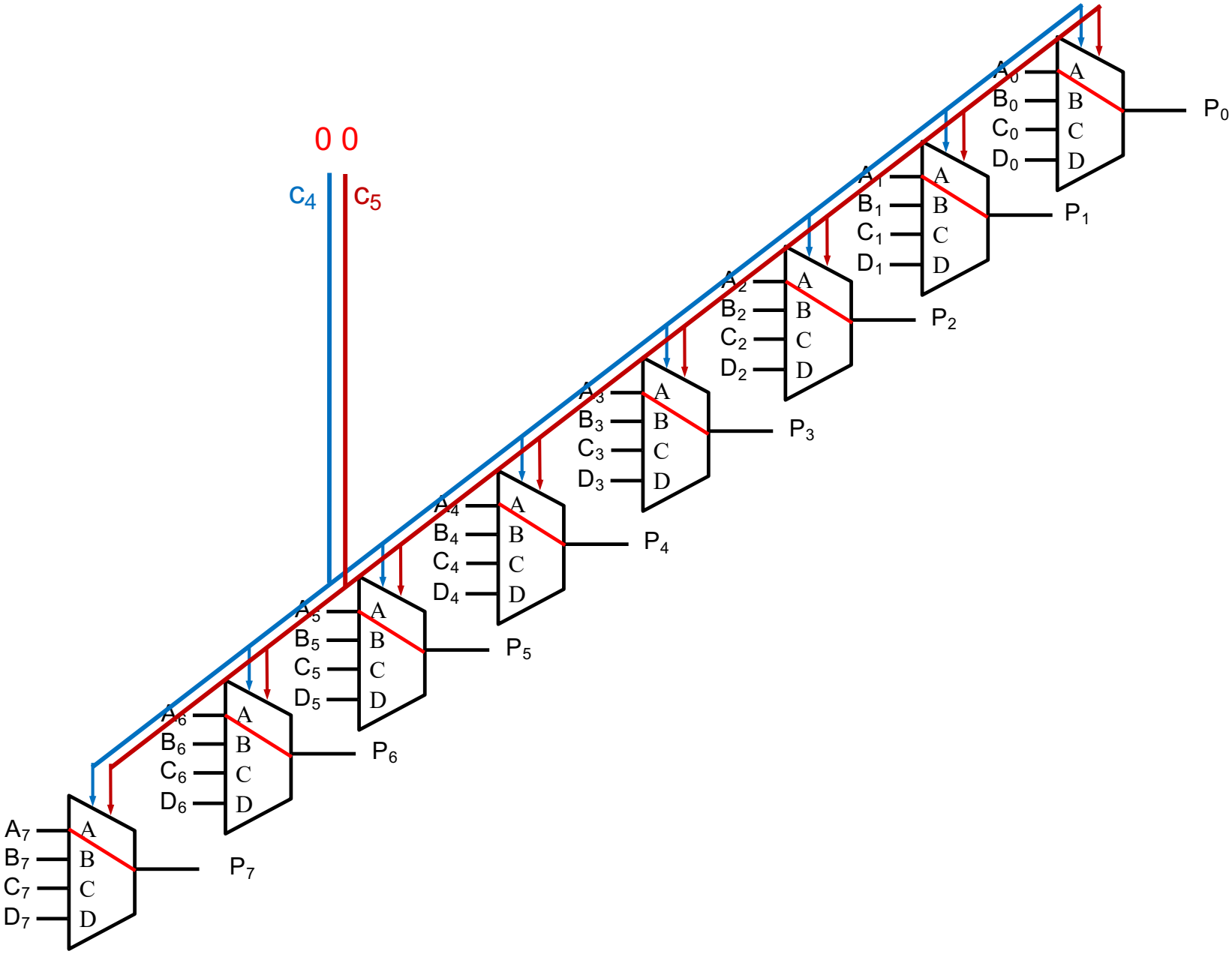

A0
B0
C0
D0
A
B
C
D
P0
A1
B1
C1
D1
P1
A2
B2
C2
D2
P2
A3
B3
C3
D3
P3
A4
B4
C4
D4
P4
A5
B5
C5
D5
P5
A6
B6
C6
D6
P6
A7
B7
C7
D7
P7

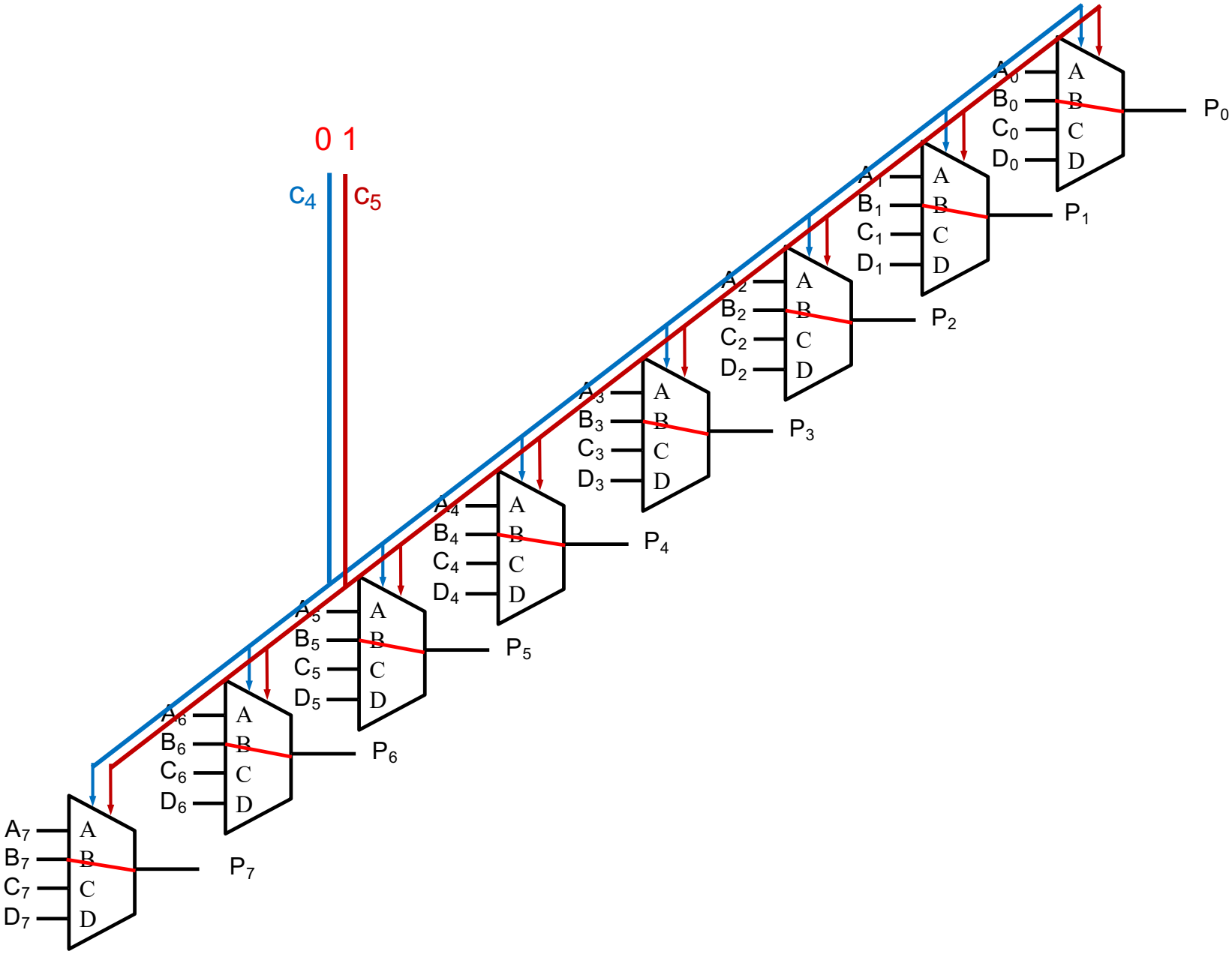

A0
B0
C0
D0
A
B
C
D
P0
A1
B1
C1
D1
P1
A2
B2
C2
D2
P2
A3
B3
C3
D3
P3
A4
B4
C4
D4
P4
A5
B5
C5
D5
P5
A6
B6
C6
D6
P6
A7
B7
C7
D7
P7
These 8 lines form the output bus
for Port_0 of the register file.

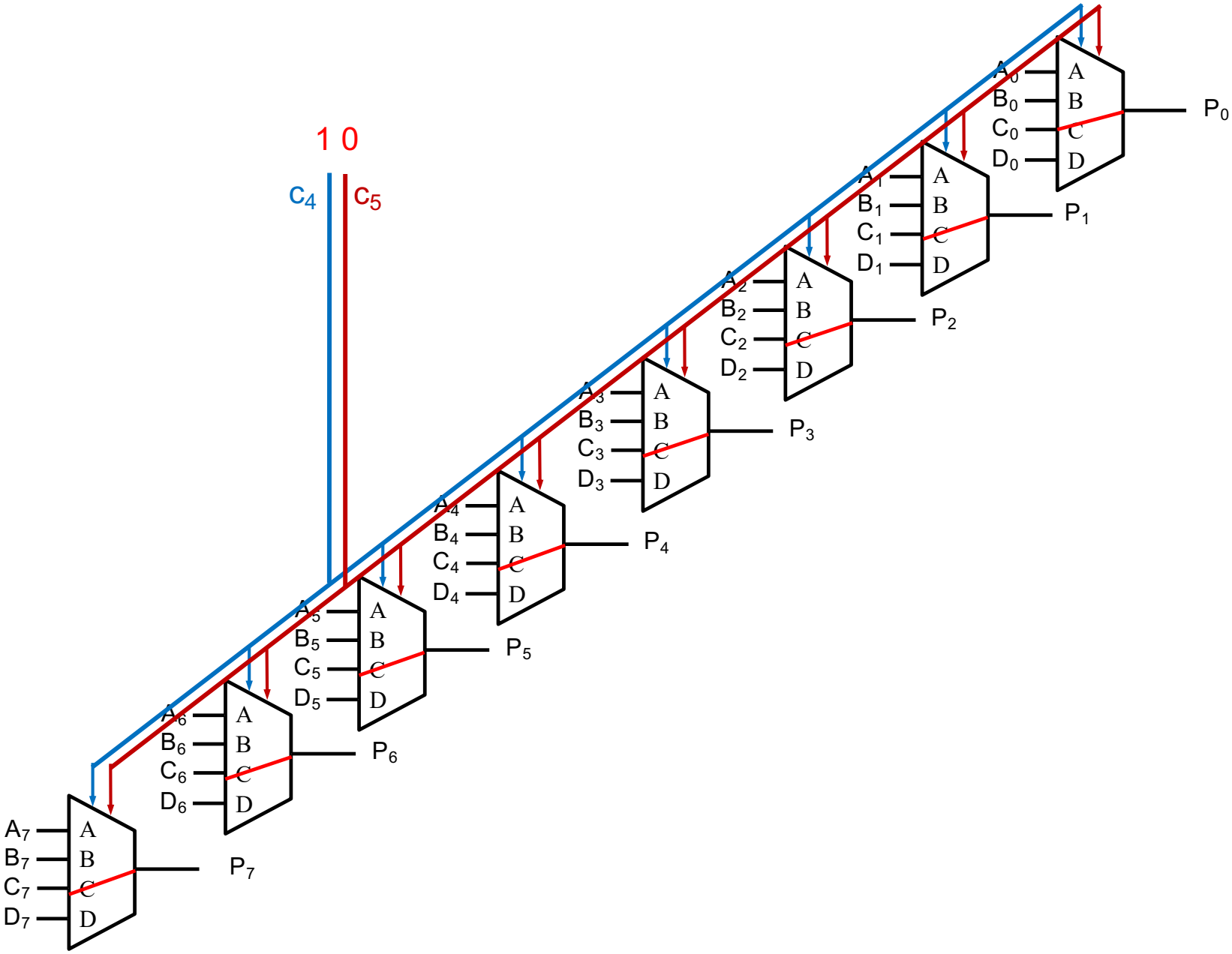

A0
B0
C0
D0
A
B
C
D
P0
A1
B1
C1
D1
P1
A2
B2
C2
D2
P2
A3
B3
C3
D3
P3
A4
B4
C4
D4
P4
A5
B5
C5
D5
P5
A6
B6
C6
D6
P6
A7
B7
C7
D7
P7

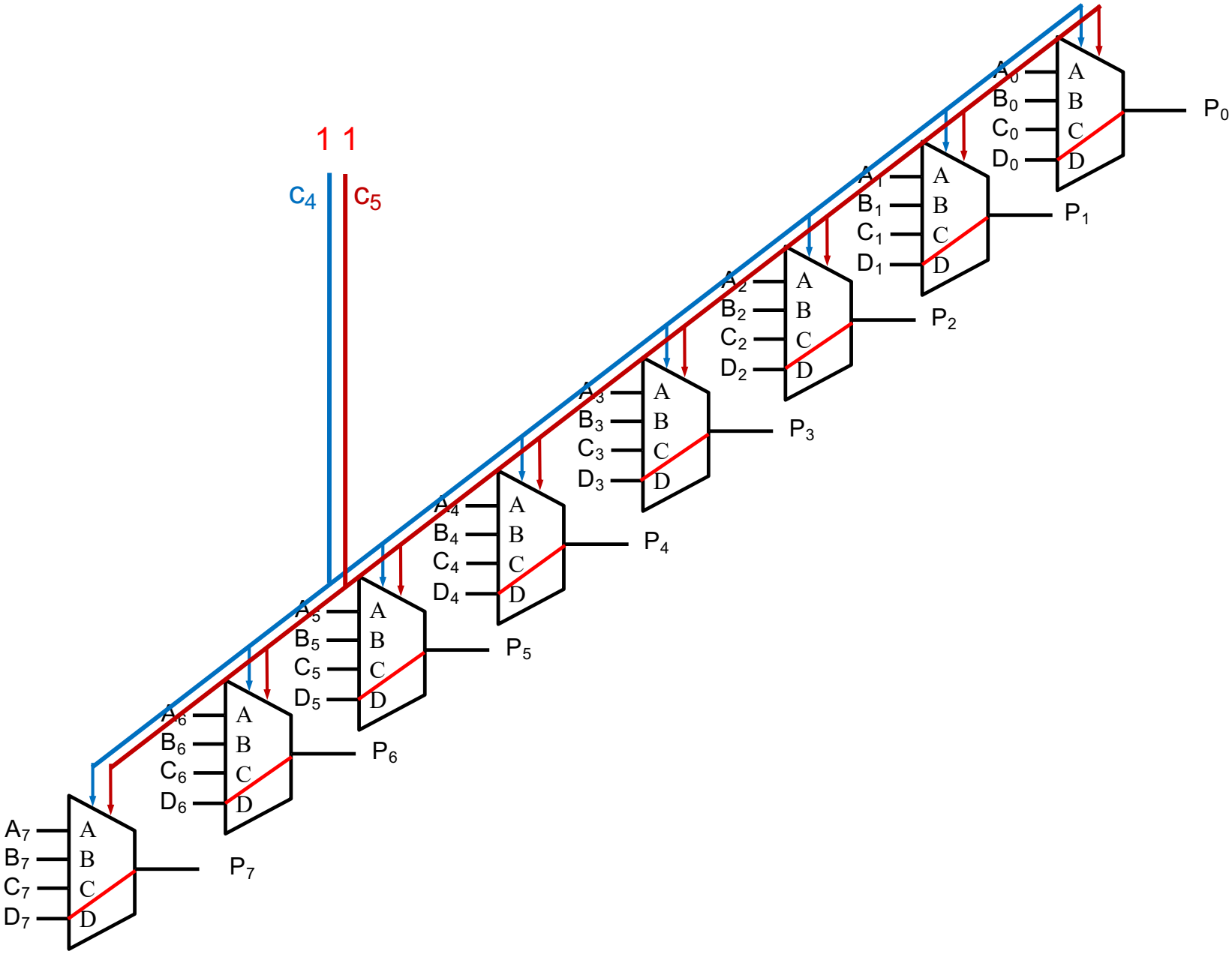

A0
B0
C0
D0
A
B
C
D
P0
A1
B1
C1
D1
P1
A2
B2
C2
D2
P2
A3
B3
C3
D3
P3
A4
B4
C4
D4
P4
A5
B5
C5
D5
P5
A6
B6
C6
D6
P6
A7
B7
C7
D7
P7

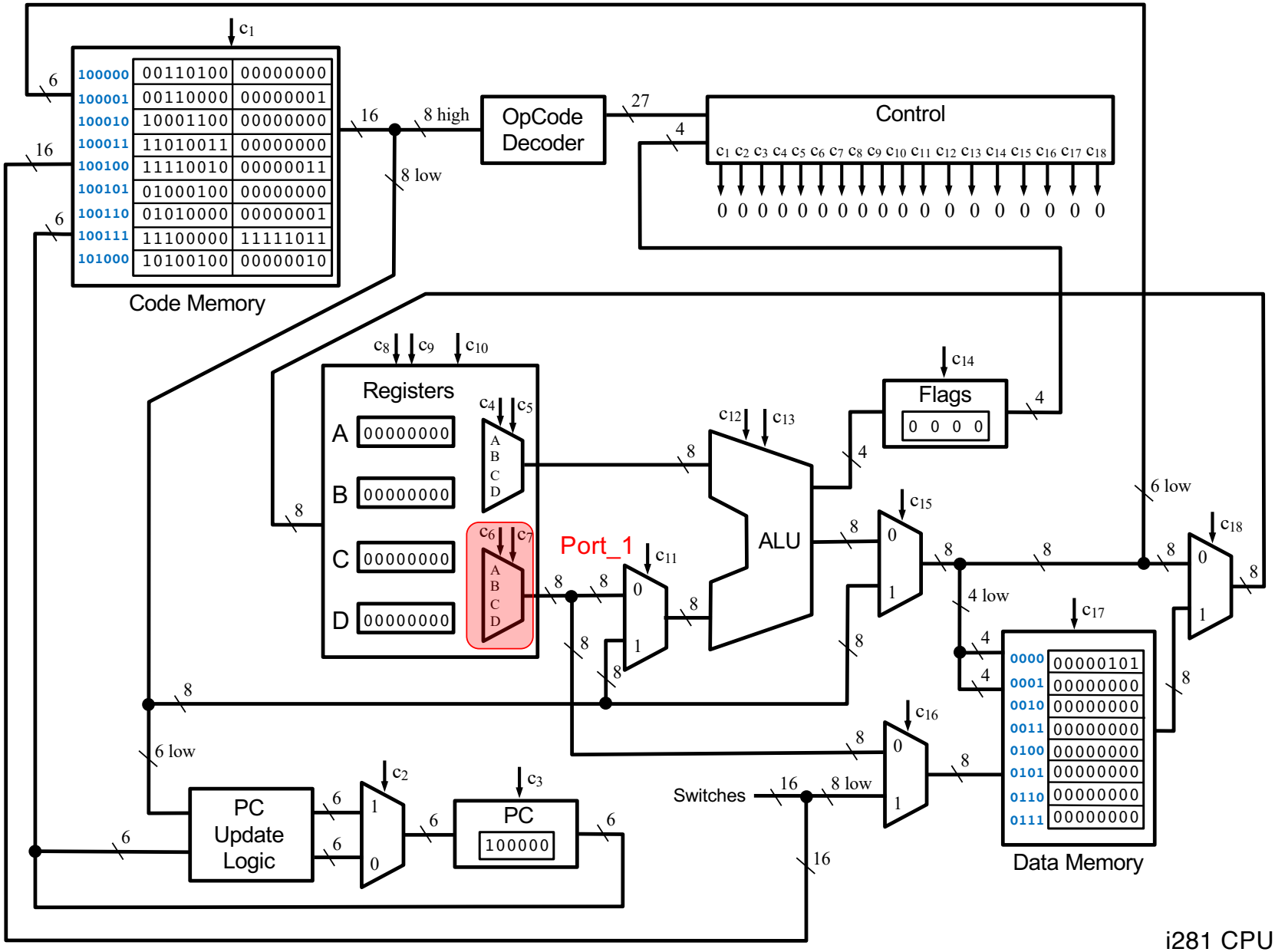

C4
C5
A0
B0
C0
D0
A
B
C
D
P0
A1
B1
C1
D1
P1
A2
B2
C2
D2
P2
A3
B3
C3
D3
P3
A4
B4
C4
D4
P4
A5
B5
C5
D5
P5
A6
B6
C6
D6
P6
A7
B7
C7
D7
P7


0 0
c_4
c_5
A_0
B_0
C_0
D_0
P_0
A_1
B_1
C_1
D_1
P_1
A_2
B_2
C_2
D_2
P_2
A_3
B_3
C_3
D_3
P_3
A_4
B_4
C_4
D_4
P_4
A_5
B_5
C_5
D_5
P_5
A_6
B_6
C_6
D_6
P_6
A_7
B_7
C_7
D_7
P_7
A
B
C
D

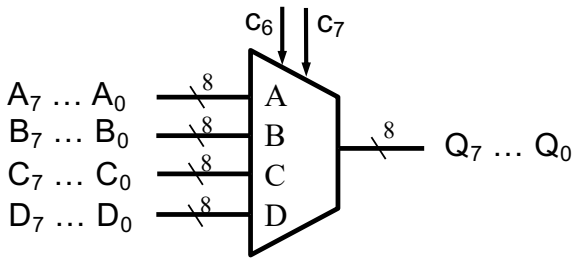

0 1
c_4
c_5
A_0
B_0
C_0
D_0
A
B
C
D
P_0
A_1
B_1
C_1
D_1
P_1
A_2
B_2
C_2
D_2
P_2
A_3
B_3
C_3
D_3
P_3
A_4
B_4
C_4
D_4
P_4
A_5
B_5
C_5
D_5
P_5
A_6
B_6
C_6
D_6
P_6
A_7
B_7
C_7
D_7
P_7

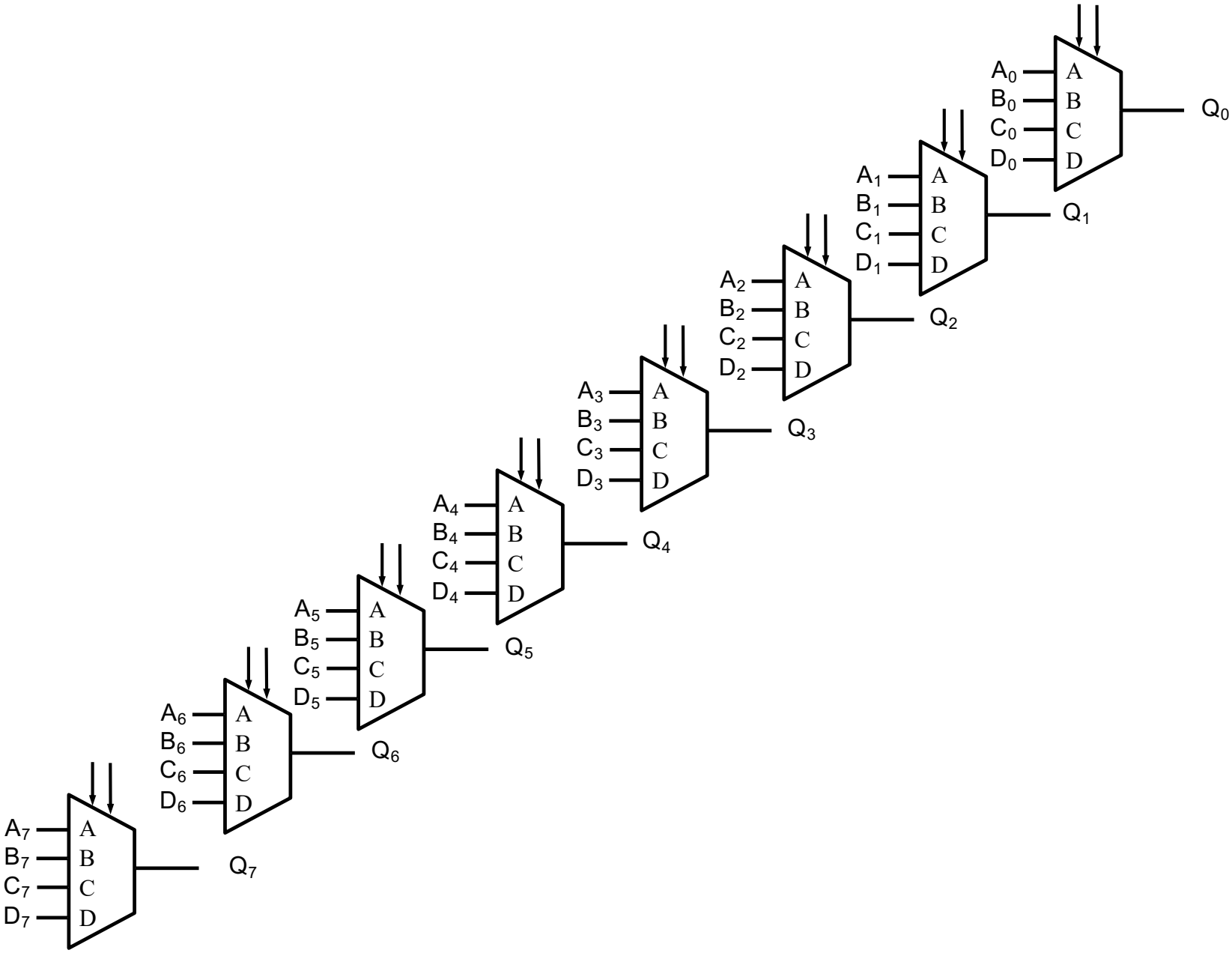

1 0
C4
C5
A7
B7
C7
D7
A
B
C
D
P7
A6
B6
C6
D6
P6
A5
B5
C5
D5
P5
A4
B4
C4
D4
P4
A3
B3
C3
D3
P3
A2
B2
C2
D2
P2
A1
B1
C1
D1
P1
A0
B0
C0
D0
P0

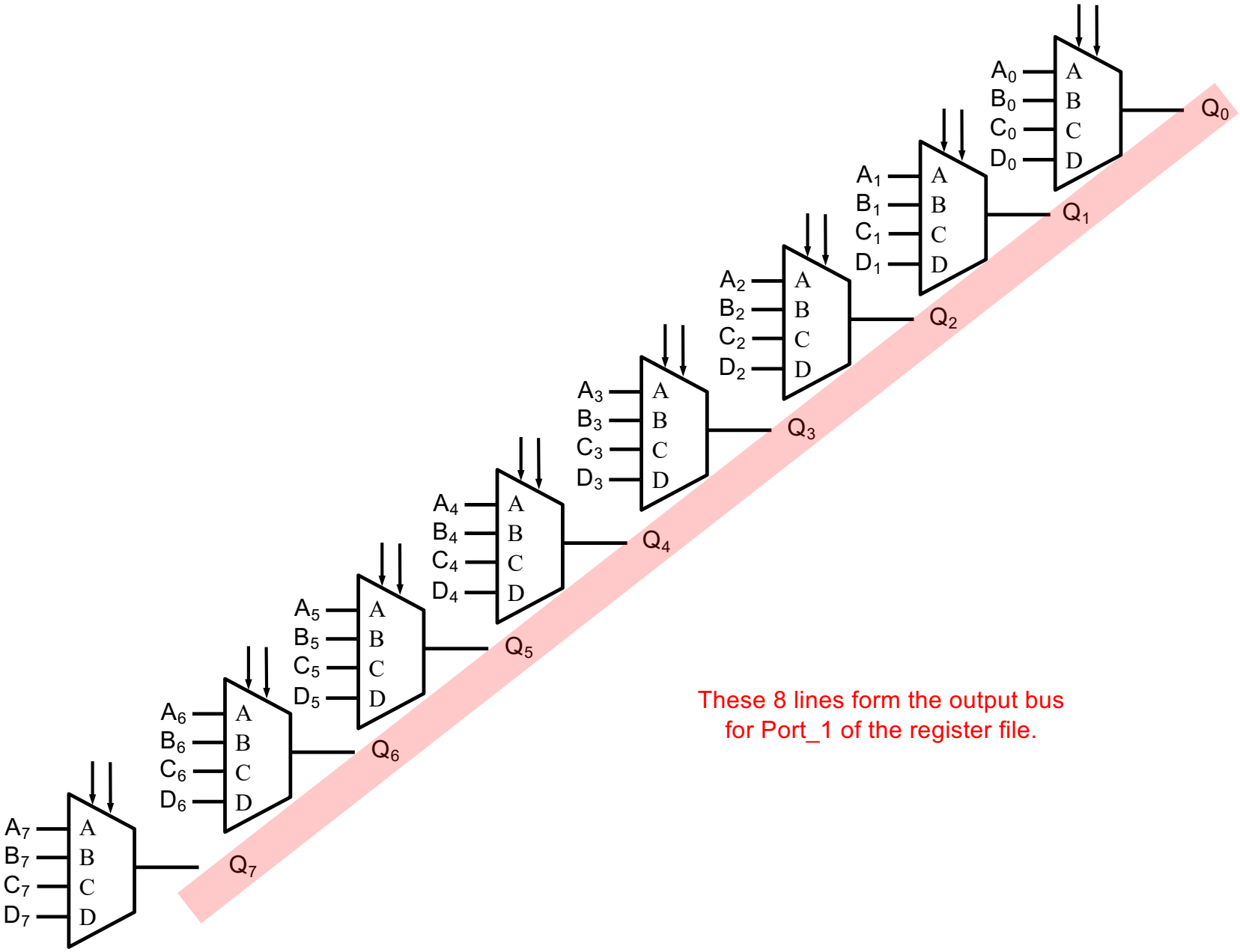

1 1
C4
C5
A7
B7
C7
D7
A
B
C
D
P7
A6
B6
C6
D6
P6
A5
B5
C5
D5
P5
A4
B4
C4
D4
P4
A3
B3
C3
D3
P3
A2
B2
C2
D2
P2
A1
B1
C1
D1
P1
A0
B0
C0
D0
P0

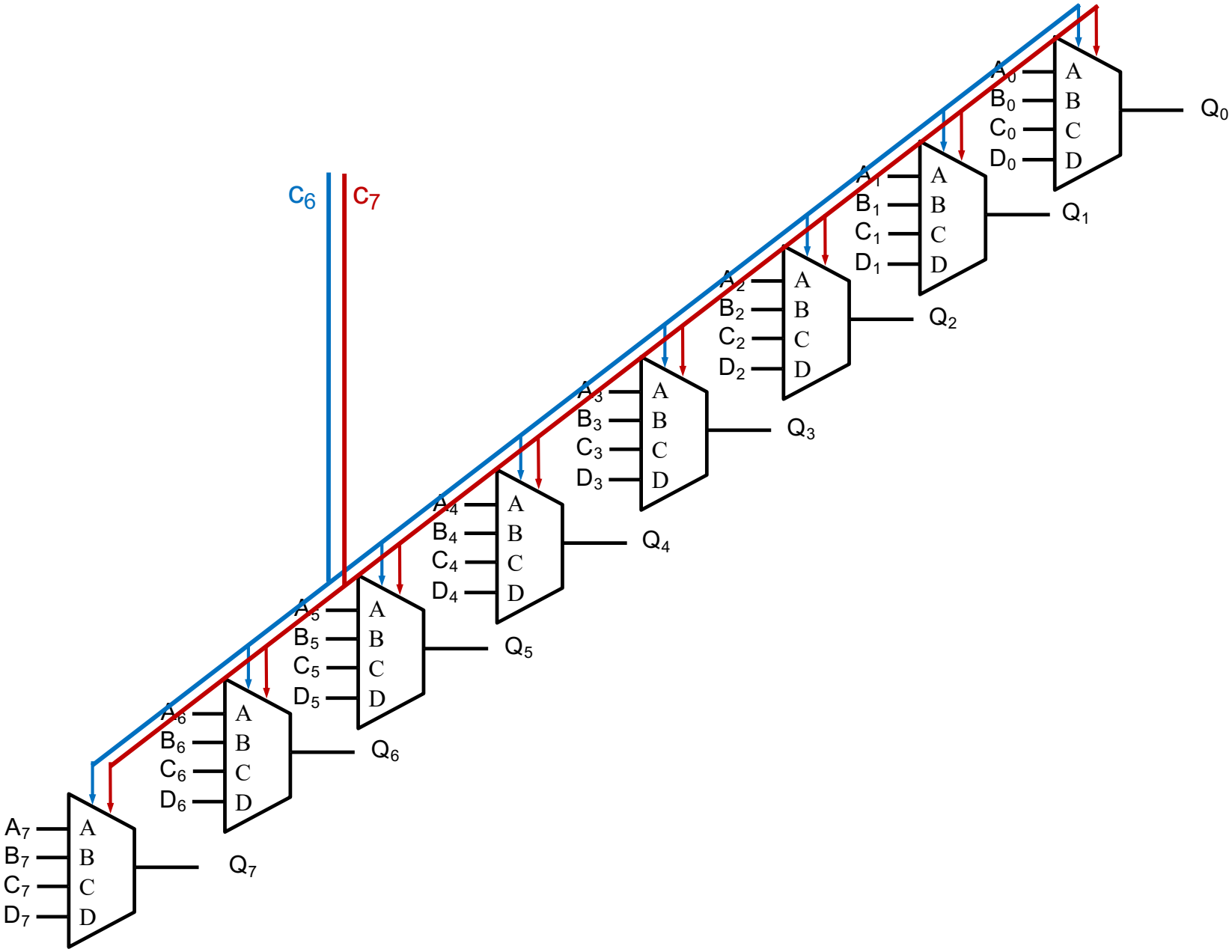

c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c12 c13
ALU
c14
Flags
0 0 0 0
4
c15
Port_1
c11
0
1
8
6 low
c18
4 low
c17
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
c16
Switches
16
8 low
6 low
c2
c3
PC Update Logic
PC
100000
6
16
i281 CPU

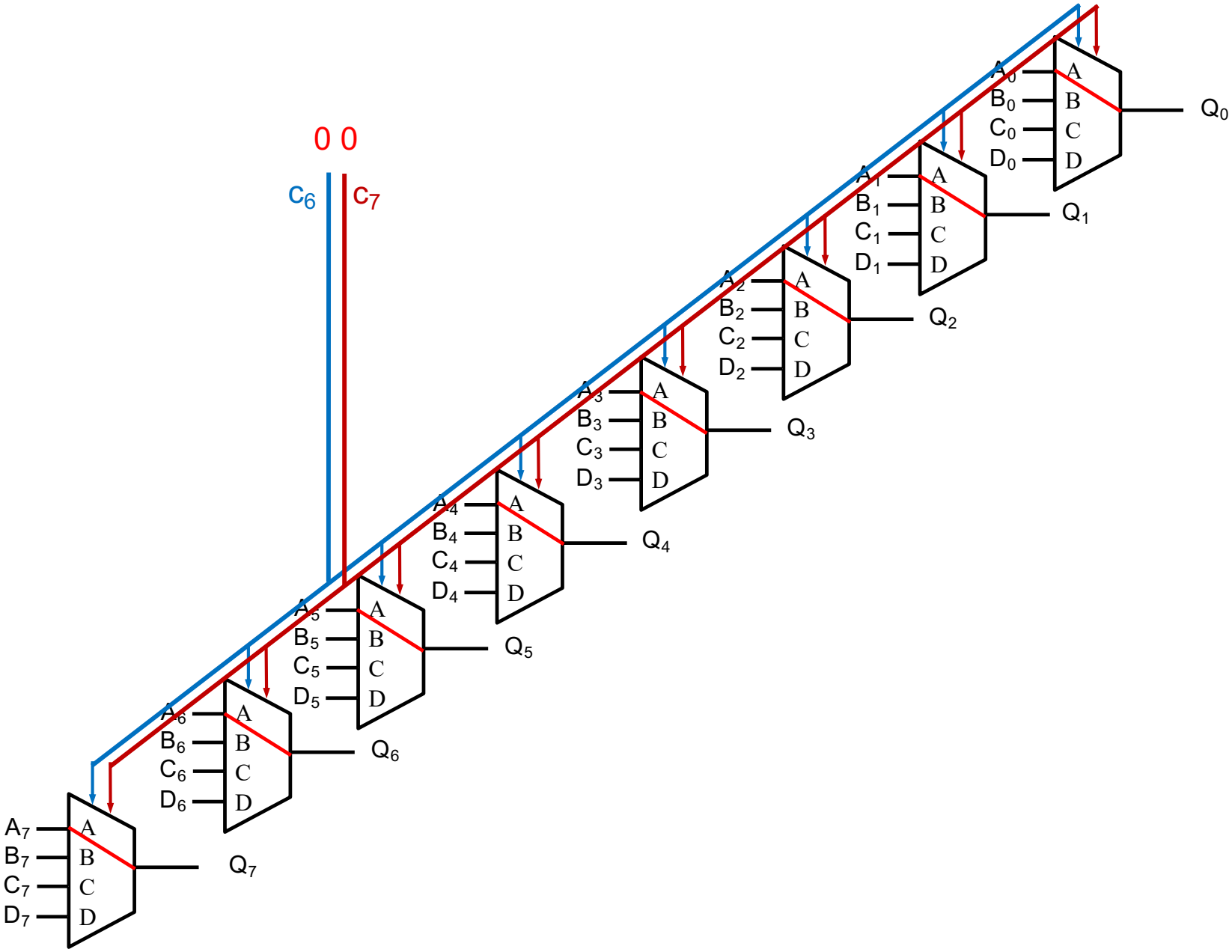
4-to-1 Bus Multiplexer (with 8-bit lines)

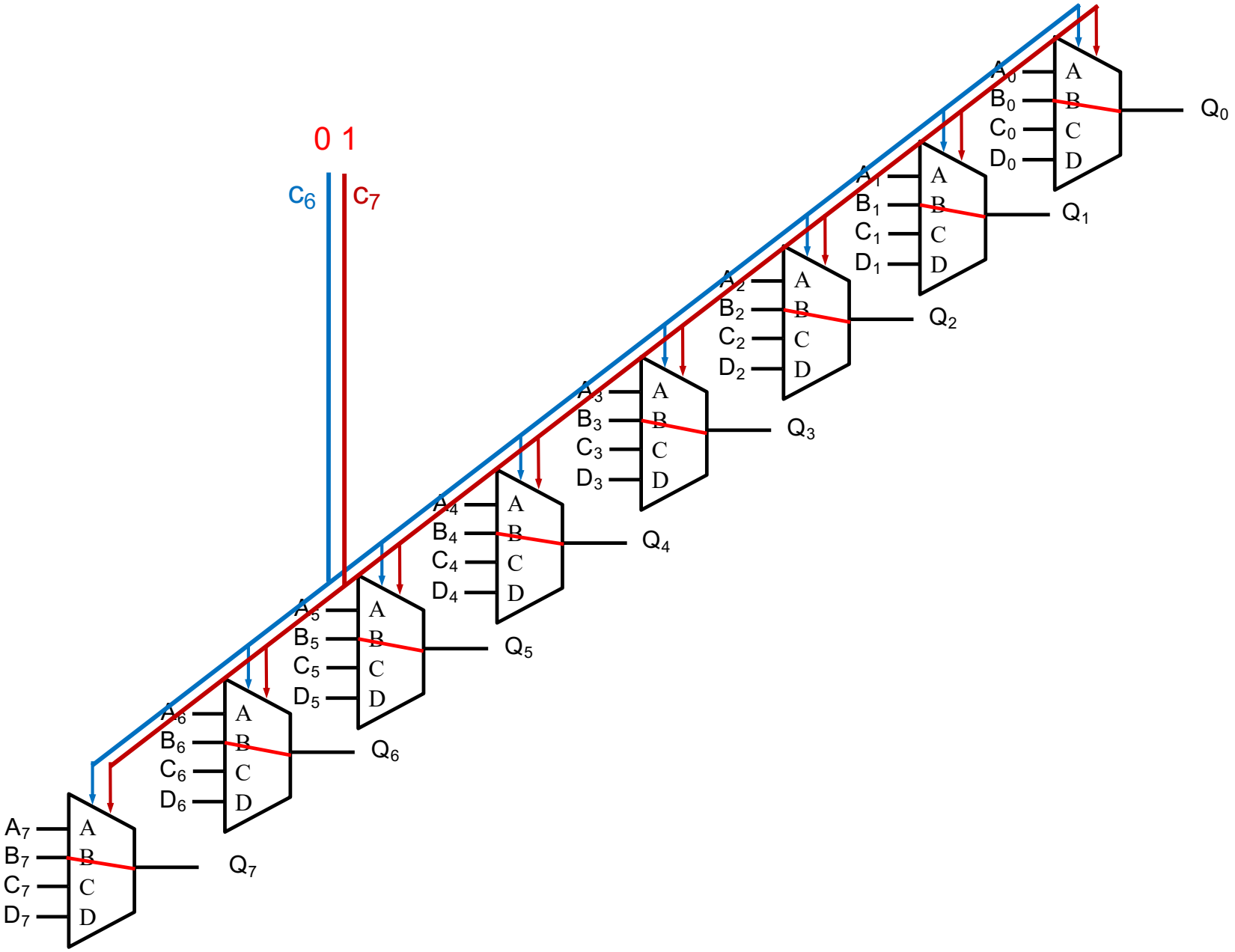


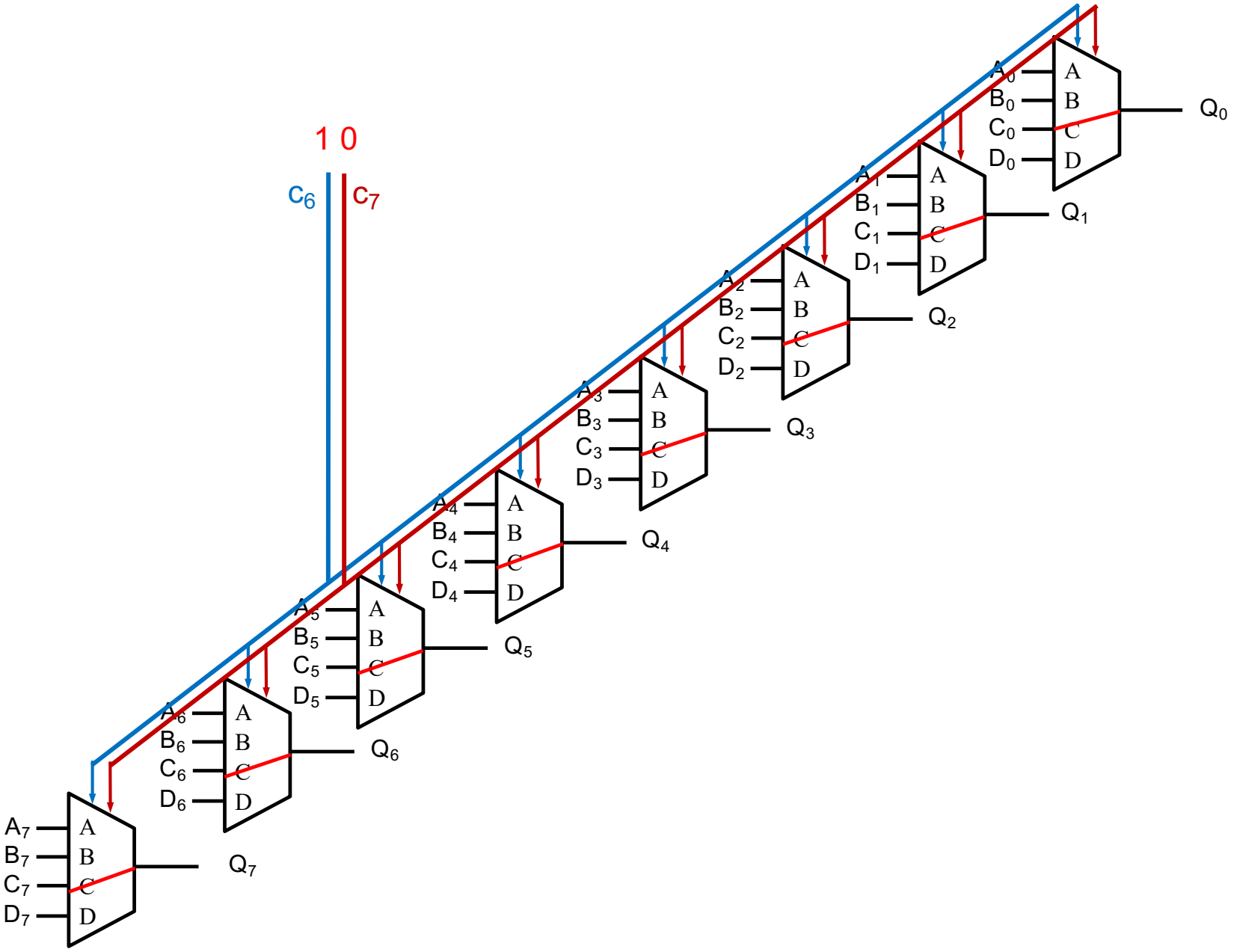

A0
B0
C0
D0
A
B
C
D
Q0
A1
B1
C1
D1
Q1
A2
B2
C2
D2
Q2
A3
B3
C3
D3
Q3
A4
B4
C4
D4
Q4
A5
B5
C5
D5
Q5
A6
B6
C6
D6
Q6
A7
B7
C7
D7
Q7

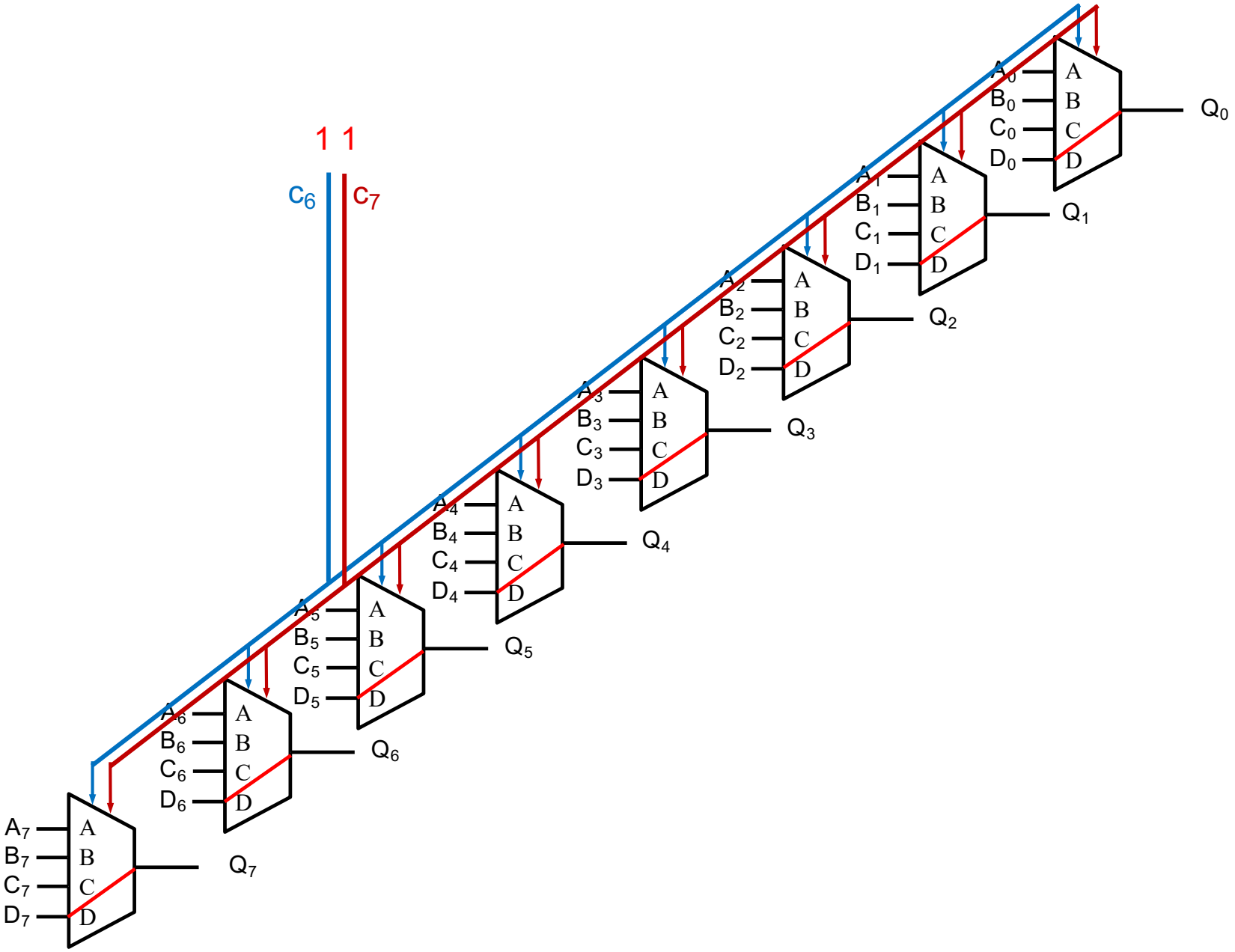

A0
B0
C0
D0
A
B
C
D
Q0
A1
B1
C1
D1
Q1
A2
B2
C2
D2
Q2
A3
B3
C3
D3
Q3
A4
B4
C4
D4
Q4
A5
B5
C5
D5
Q5
A6
B6
C6
D6
Q6
A7
B7
C7
D7
Q7
These 8 lines form the output bus
for Port_1 of the register file.

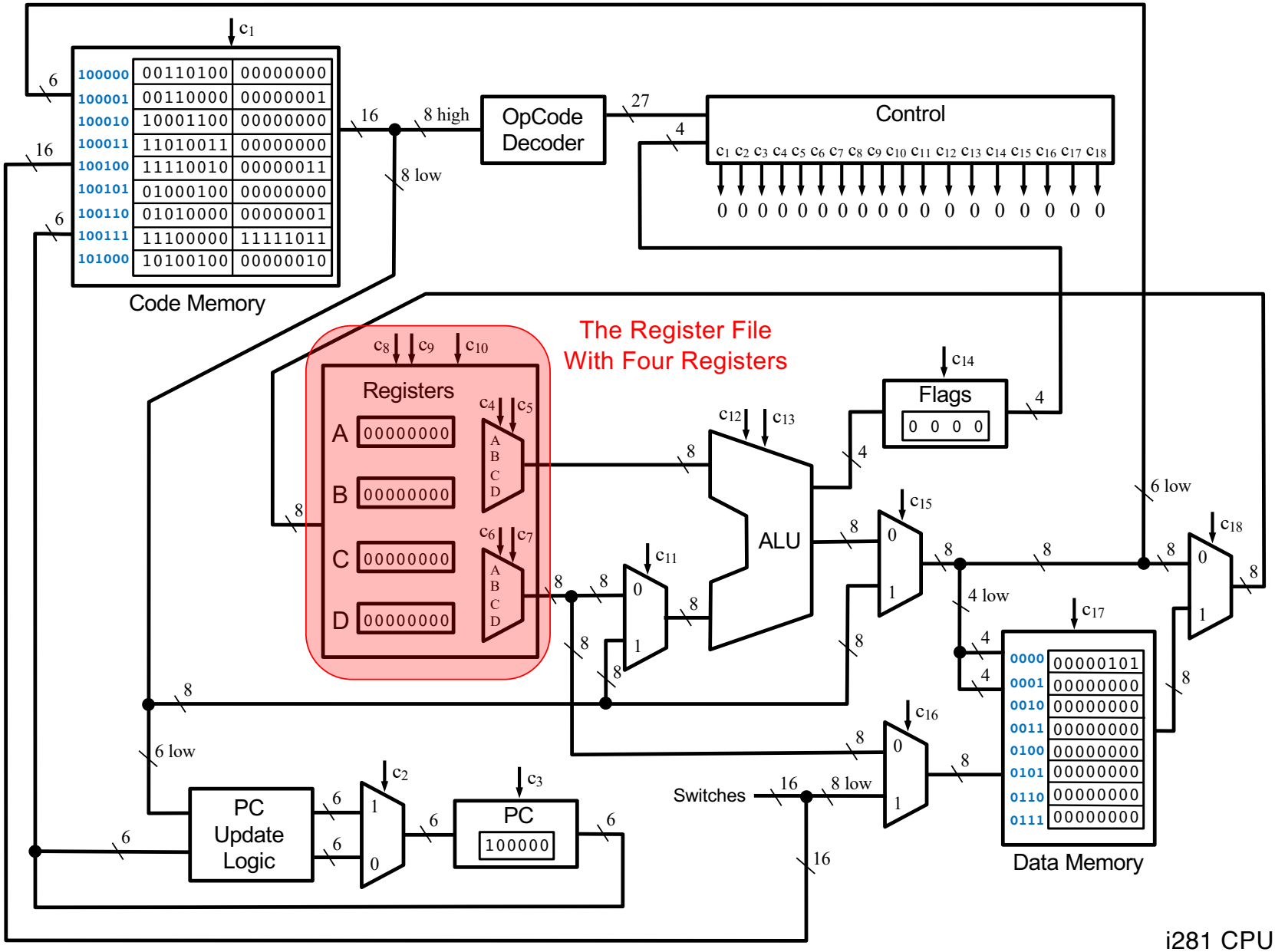

C6
C7
A7
B7
C7
D7
A
B
C
D
Q7
A6
B6
C6
D6
Q6
A5
B5
C5
D5
Q5
A4
B4
C4
D4
Q4
A3
B3
C3
D3
Q3
A2
B2
C2
D2
Q2
A1
B1
C1
D1
Q1
A0
B0
C0
D0
Q0


0 0
C6
C7
A7
B7
C7
D7
A
B
C
D
Q7
A6
B6
C6
D6
Q6
A5
B5
C5
D5
Q5
A4
B4
C4
D4
Q4
A3
B3
C3
D3
Q3
A2
B2
C2
D2
Q2
A1
B1
C1
D1
Q1
A0
B0
C0
D0
Q0


0 1
c6
c7
A7
B7
C7
D7
A
B
C
D
Q7
A6
B6
C6
D6
Q6
A5
B5
C5
D5
Q5
A4
B4
C4
D4
Q4
A3
B3
C3
D3
Q3
A2
B2
C2
D2
Q2
A1
B1
C1
D1
Q1
A0
B0
C0
D0
Q0

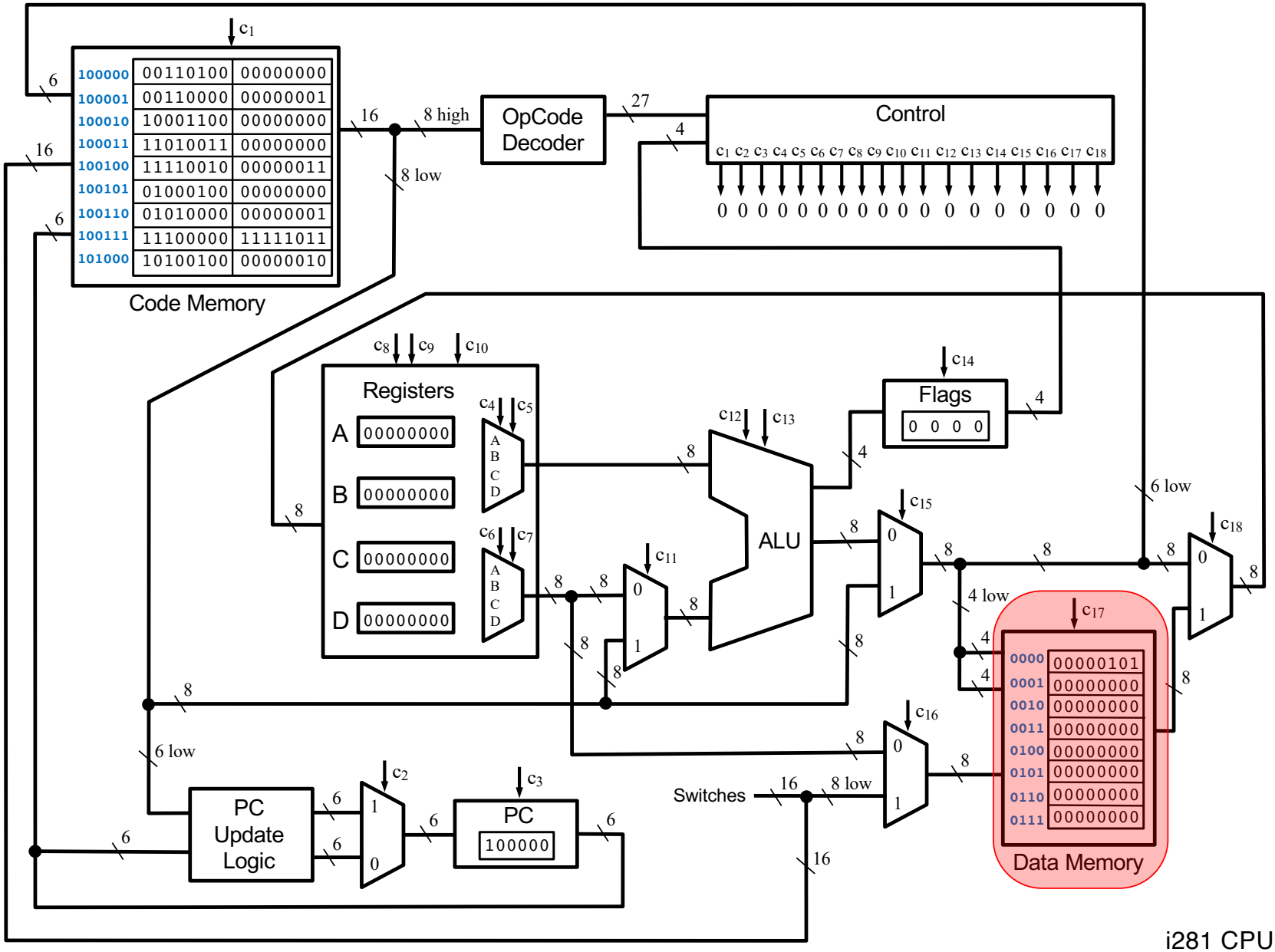

1 0
c_6
c_7
A_7
B_7
C_7
D_7
Q_7
A_6
B_6
C_6
D_6
Q_6
A_5
B_5
C_5
D_5
Q_5
A_4
B_4
C_4
D_4
Q_4
A_3
B_3
C_3
D_3
Q_3
A_2
B_2
C_2
D_2
Q_2
A_1
B_1
C_1
D_1
Q_1
A_0
B_0
C_0
D_0
Q_0
A
B
C
D

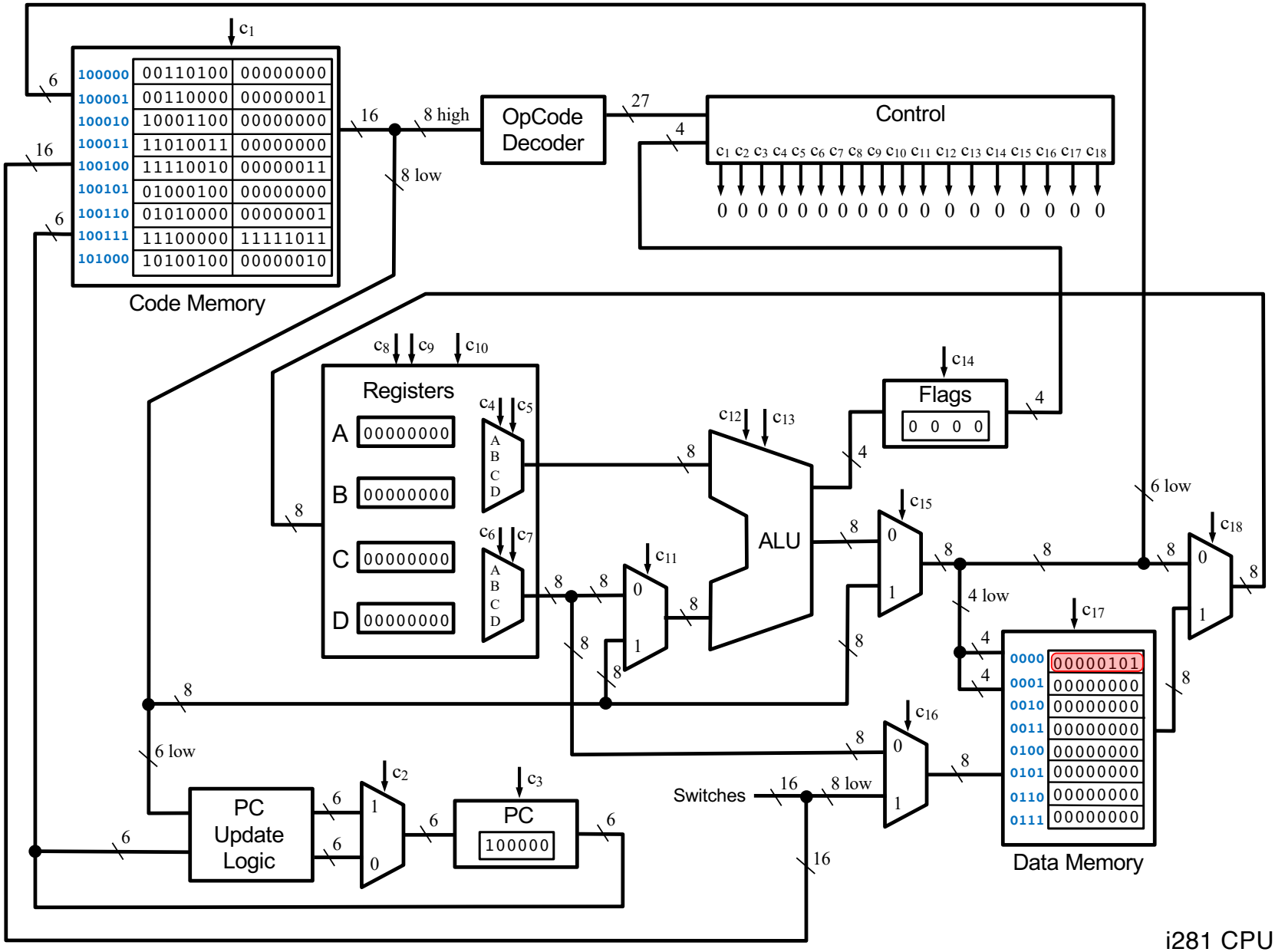

1 1
c_6
c_7
A_7
B_7
C_7
D_7
A_6
B_6
C_6
D_6
A_5
B_5
C_5
D_5
A_4
B_4
C_4
D_4
A_3
B_3
C_3
D_3
A_2
B_2
C_2
D_2
A_1
B_1
C_1
D_1
A_0
B_0
C_0
D_0
A
B
C
D
Q_7
Q_6
Q_5
Q_4
Q_3
Q_2
Q_1
Q_0


c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
The Register File
With Four Registers
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
c14
Flags
0 0 0 0
c12 c13
ALU
c15
c11
c16
c17
c18
6 low
4 low
8 low
Switches
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
PC Update Logic
c2
c3
PC
100000
i281 CPU

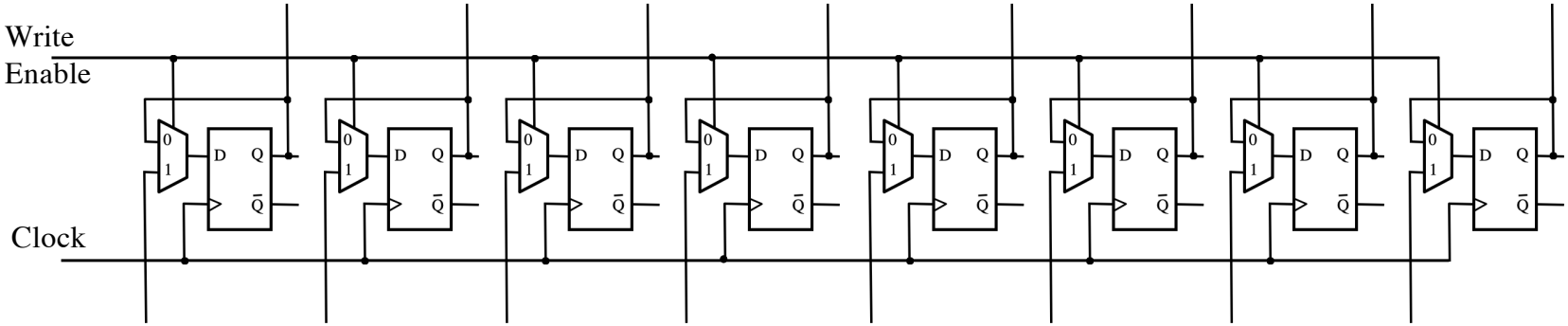
The Data Memory

(implemented as a register file with sixteen 8-bit registers that has one read port and one write port)


c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c12 c13
ALU
c14
Flags
0 0 0 0
4
c15
c11
c16
c17
c18
6 low
4 low
8 low
Switches
16
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
6 low
PC Update Logic
c2
c3
PC
100000
i281 CPU

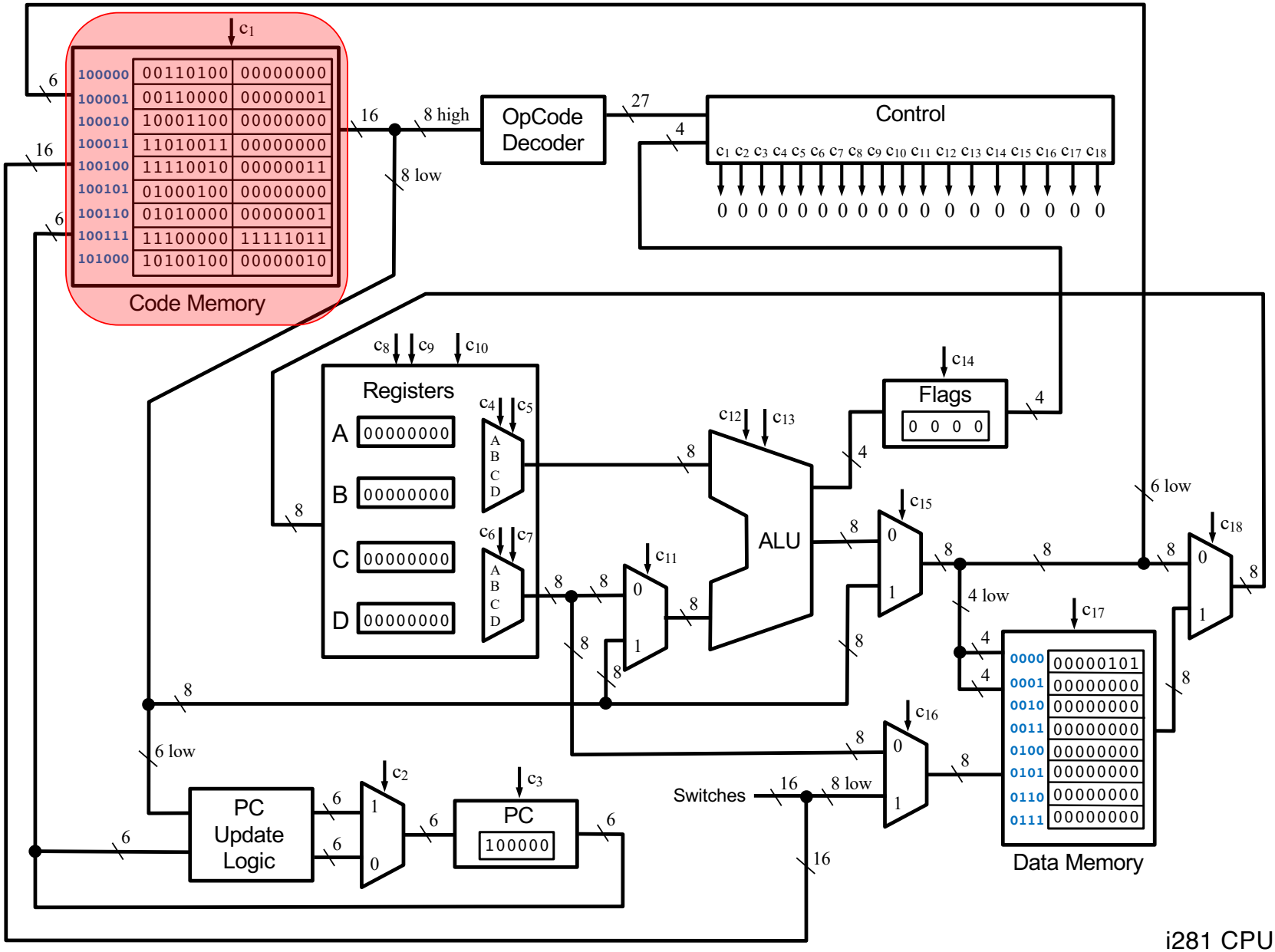

c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c12 c13
ALU
c14
Flags
0 0 0 0
4
c15
c11
c16
c17
c18
6 low
4 low
8 low
Switches
16
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
c2
c3
PC Update Logic
PC
100000
i281 CPU

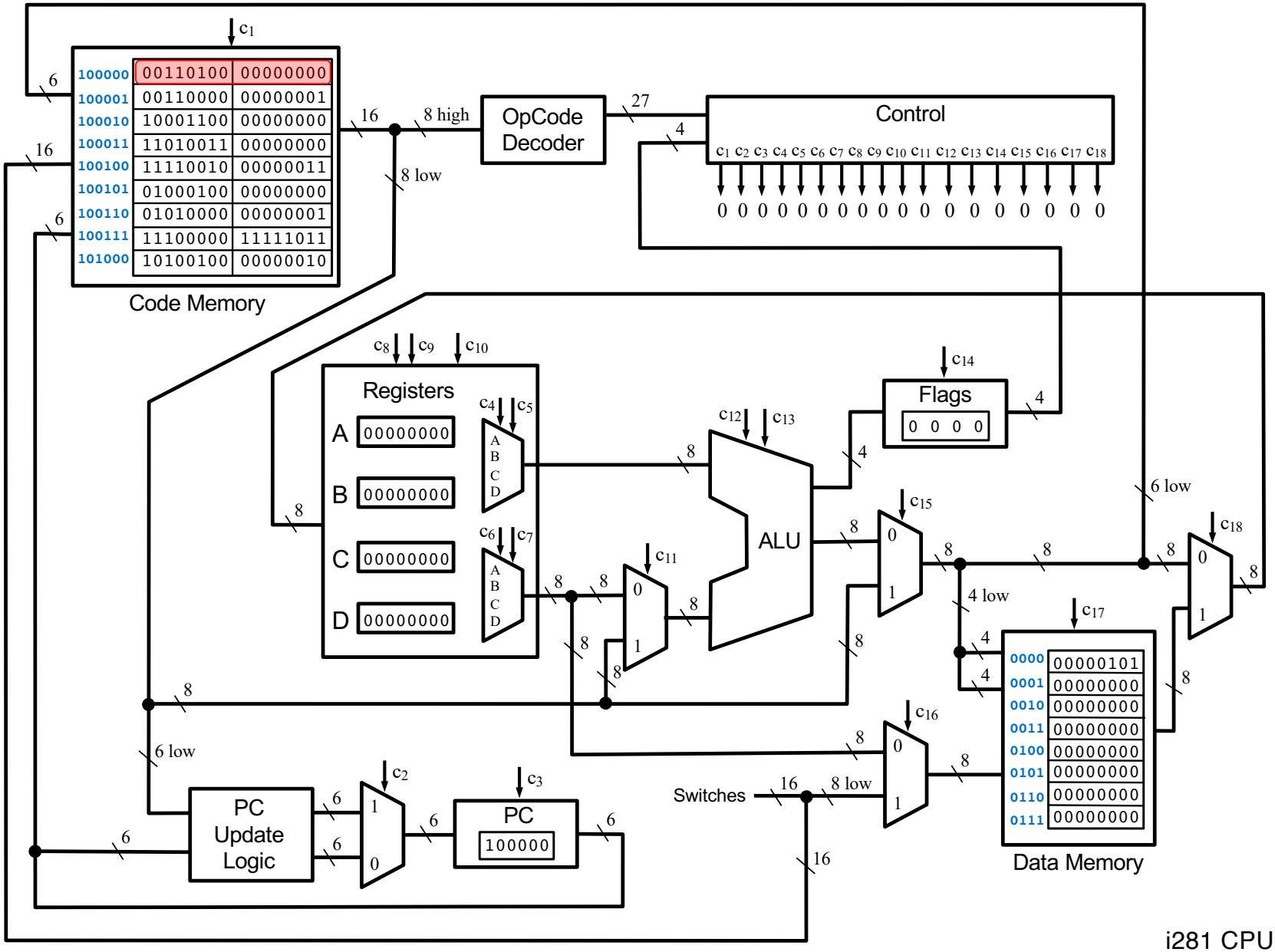
8-Bit Parallel-Access Register



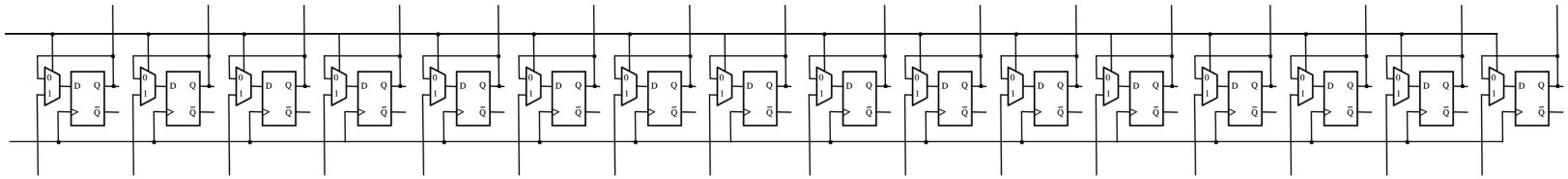
The Code Memory

**(implemented as a register file with
64 16-bit registers that has
one read port and one write port)**

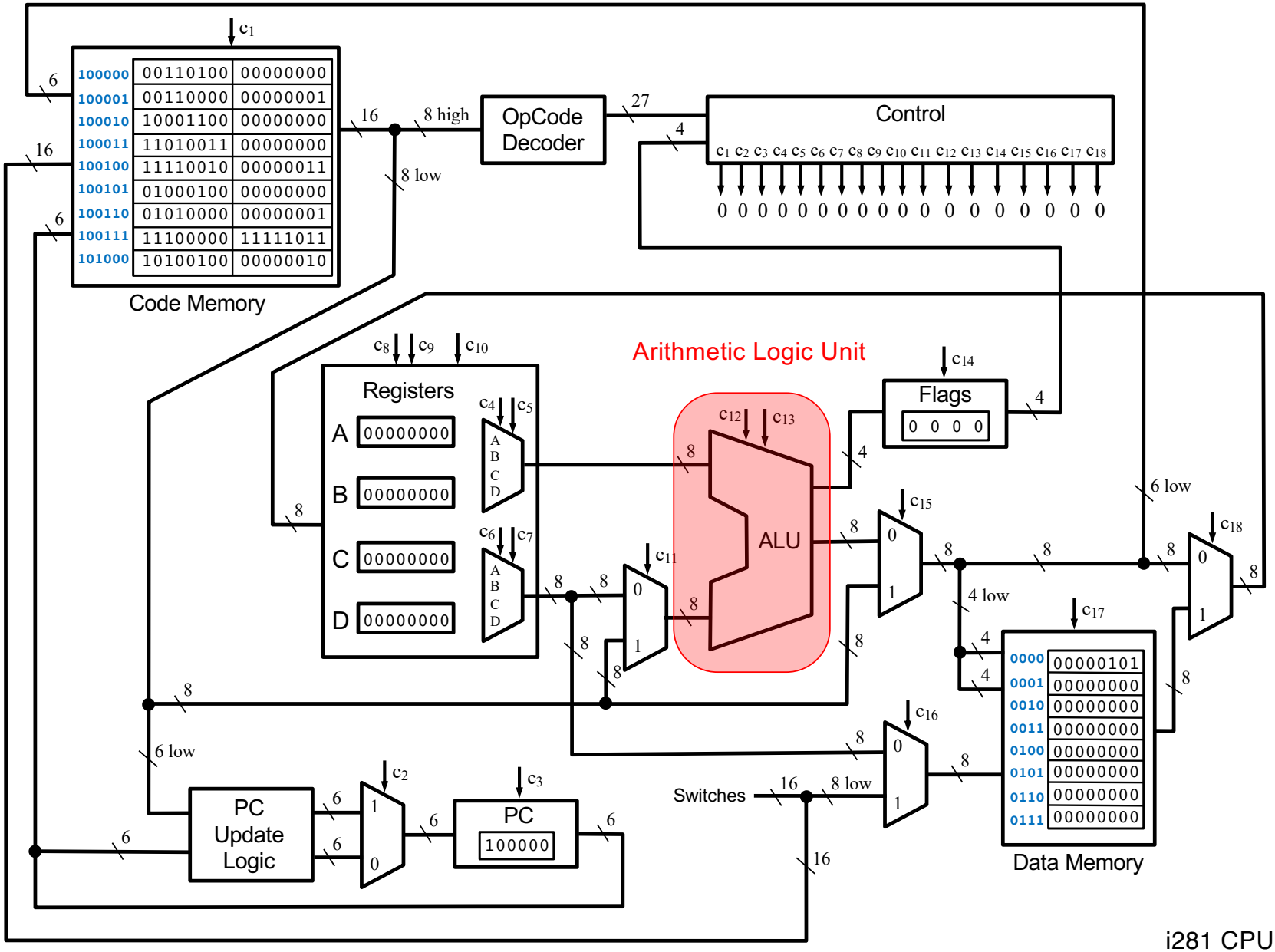

c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
c12 c13
ALU
c14
Flags
0 0 0 0
4
c15
c11
c16
c17
c18
6 low
4 low
8 low
8 high
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
Switches
c2
c3
PC Update Logic
PC
100000
i281 CPU

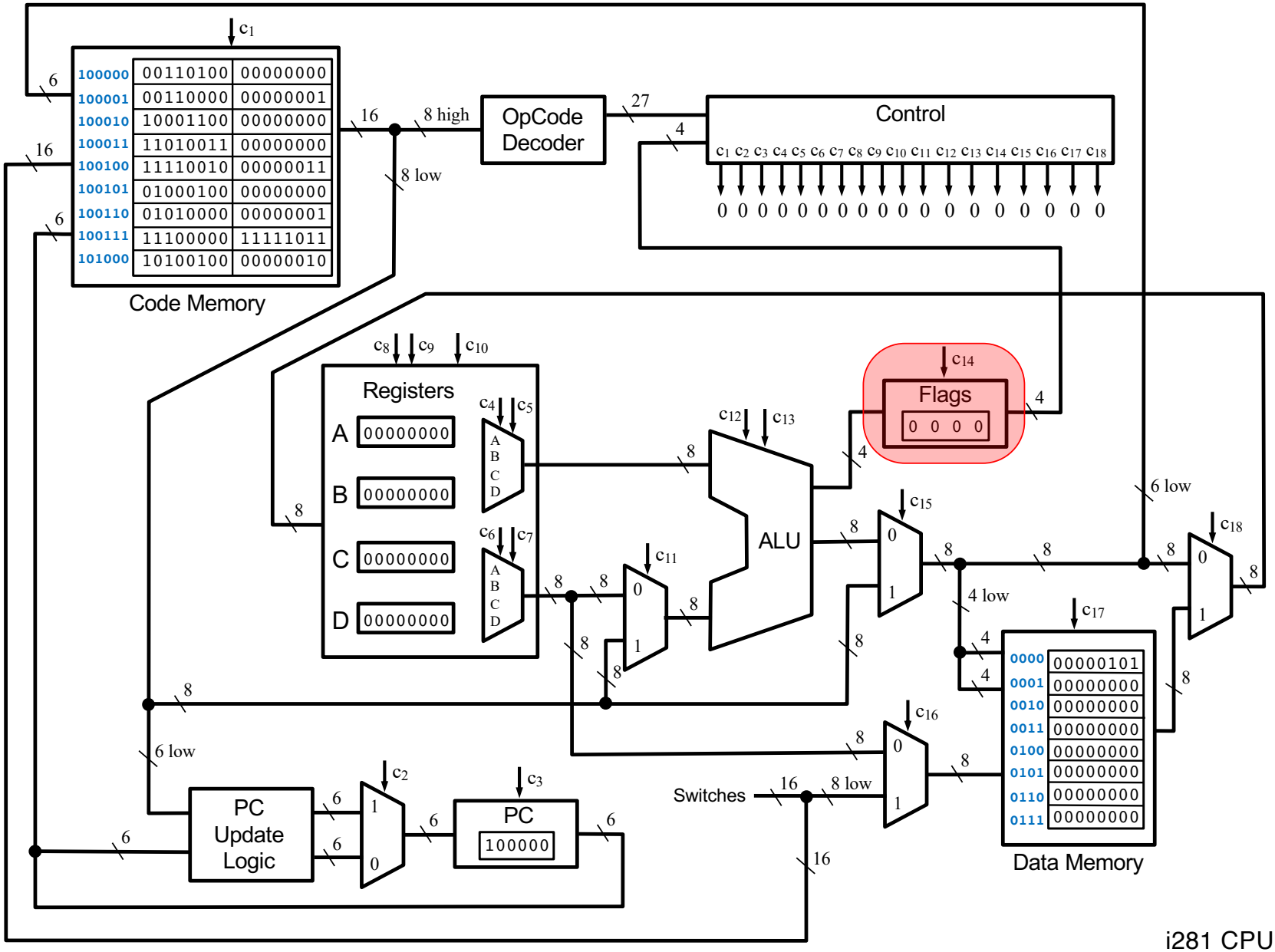

c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c14
Flags
0 0 0 0
4
c12 c13
ALU
c15
c11
c16
c17
c18
6 low
4 low
8 low
Switches
16
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
c2
c3
PC Update Logic
PC
100000
i281 CPU

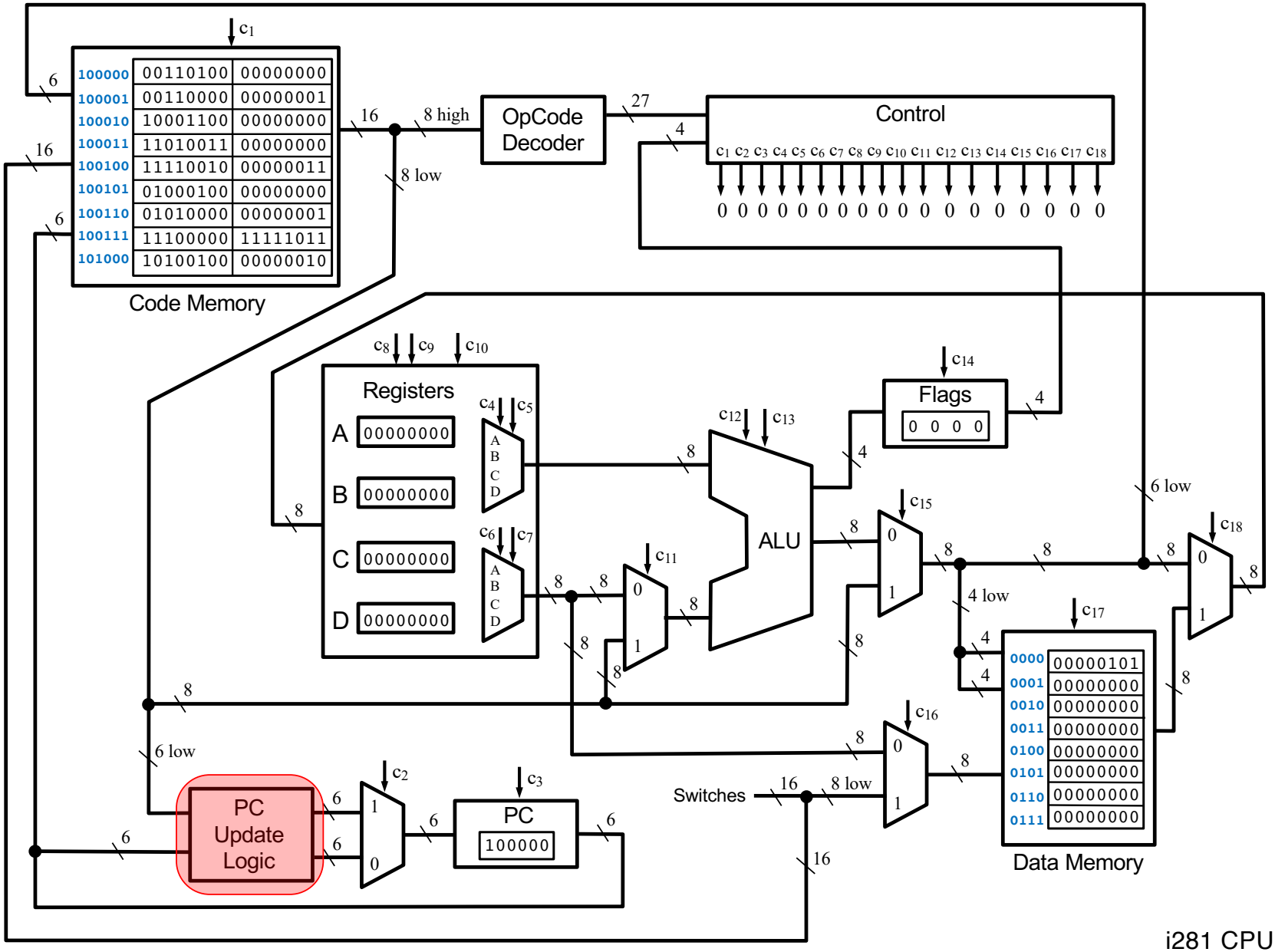
16-Bit Parallel-Access Register

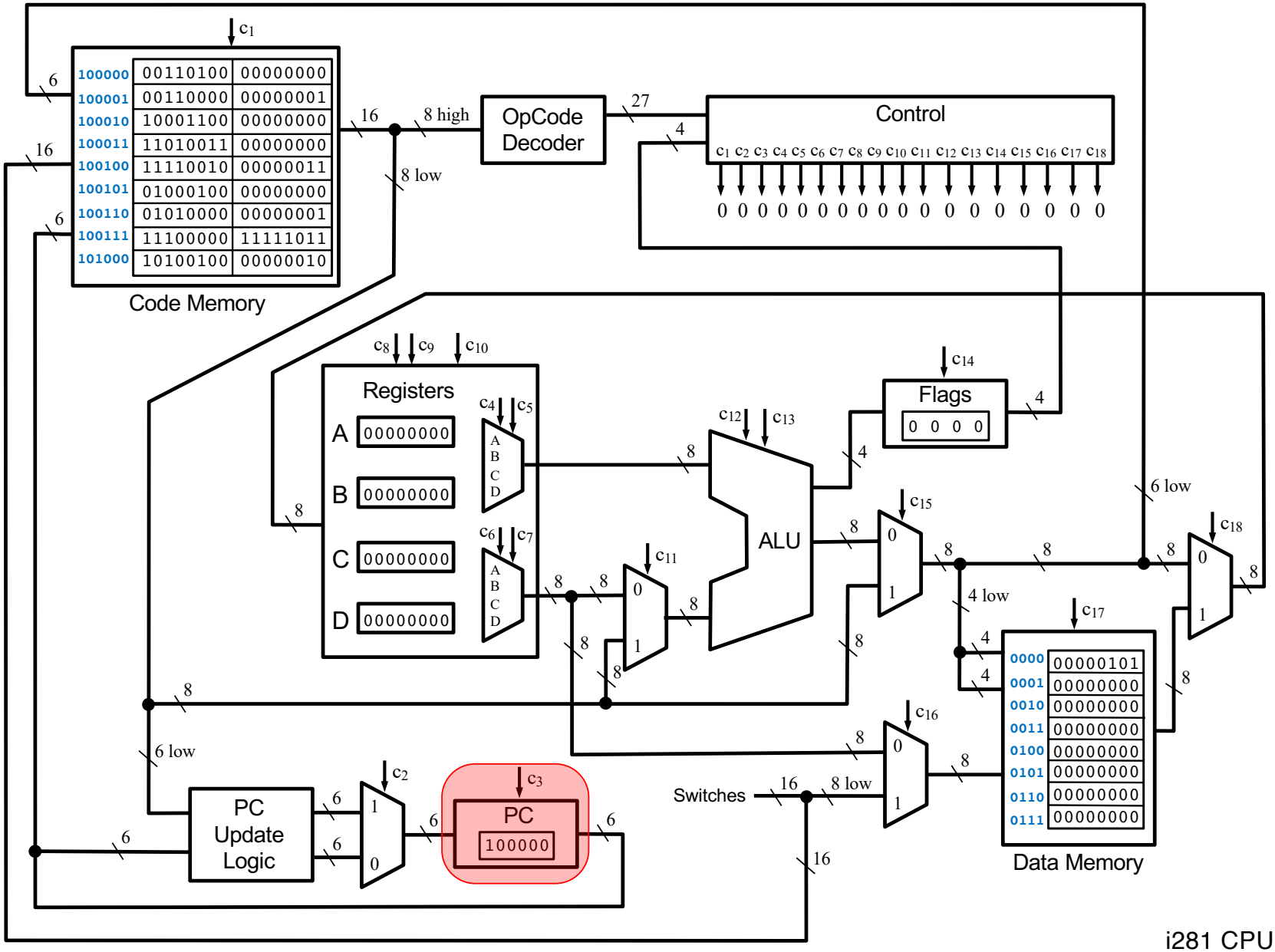


To Be Covered Later


c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Arithmetic Logic Unit
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c12 c13
ALU
c14
Flags
0 0 0 0
4
c15
c11
c16
c17
c18
6 low
4 low
8 low
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
Switches
16
PC Update Logic
c2
c3
PC
100000
i281 CPU


c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c14
Flags
0 0 0 0
4
c12 c13
ALU
c15
c11
c16
c17
c18
6 low
4 low
8 low
Switches
16
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
PC Update Logic
c2
c3
PC
100000
i281 CPU

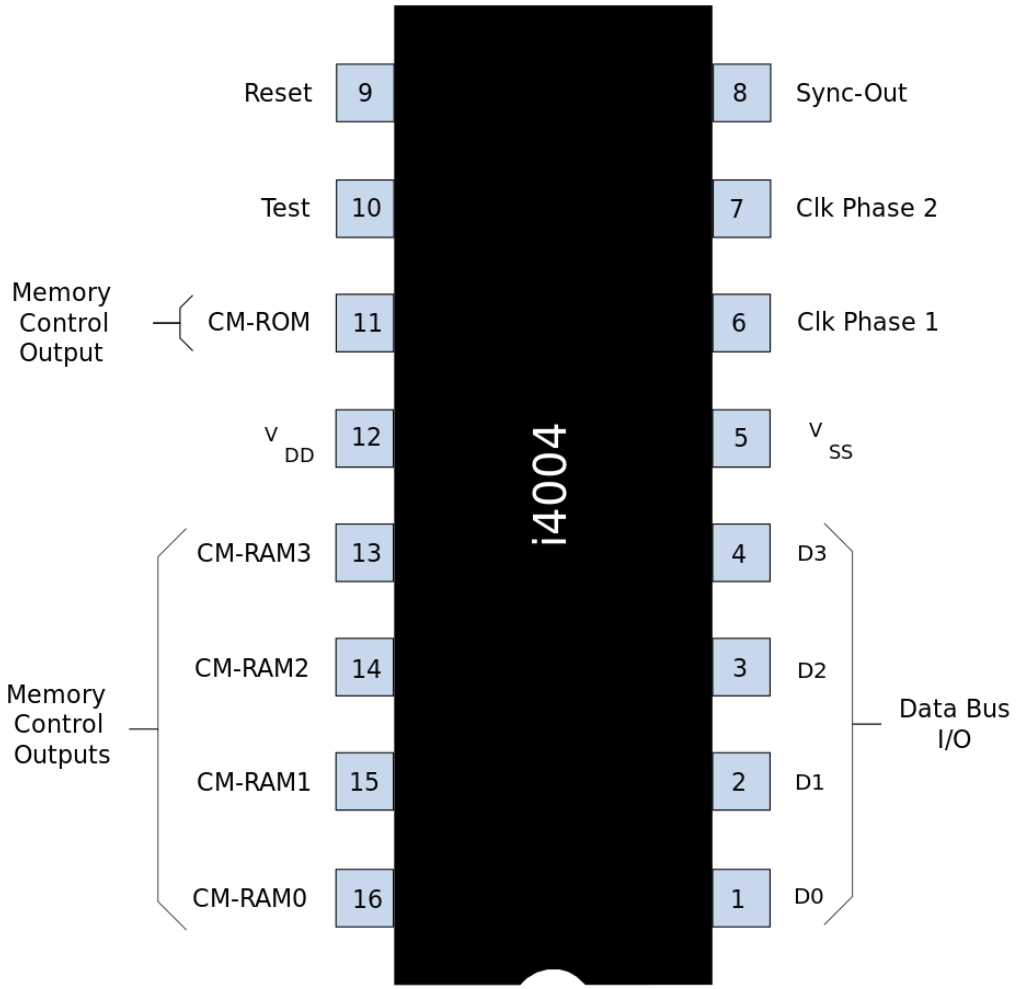

c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
6
16
6
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
8
c12 c13
ALU
c14
Flags
0 0 0 0
4
c15
c11
c16
c17
c18
6 low
4 low
8 low
Switches
16
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
PC Update Logic
c2
c3
PC
100000
i281 CPU


c1
100000 00110100 00000000
100001 00110000 00000001
100010 10001100 00000000
100011 11010011 00000000
100100 11110010 00000011
100101 01000100 00000000
100110 01010000 00000001
100111 11100000 11111011
101000 10100100 00000010
Code Memory
16
8 high
8 low
OpCode Decoder
27
4
Control
c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14 c15 c16 c17 c18
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
c8 c9 c10
Registers
A 00000000
B 00000000
C 00000000
D 00000000
c4 c5
c6 c7
A B C D
c12 c13
ALU
c14
Flags
0 0 0 0
c15
c11
c16
c17
c18
6 low
4 low
Switches
8 low
0000 00000101
0001 00000000
0010 00000000
0011 00000000
0100 00000000
0101 00000000
0110 00000000
0111 00000000
Data Memory
c2
c3
PC
100000
PC Update Logic
i281 CPU

Some Additional Topics

Examples of Some Famous Microprocessors

Intel's 4004 Chip



[http://en.wikipedia.org/wiki/Intel_4004]

Technical specifications

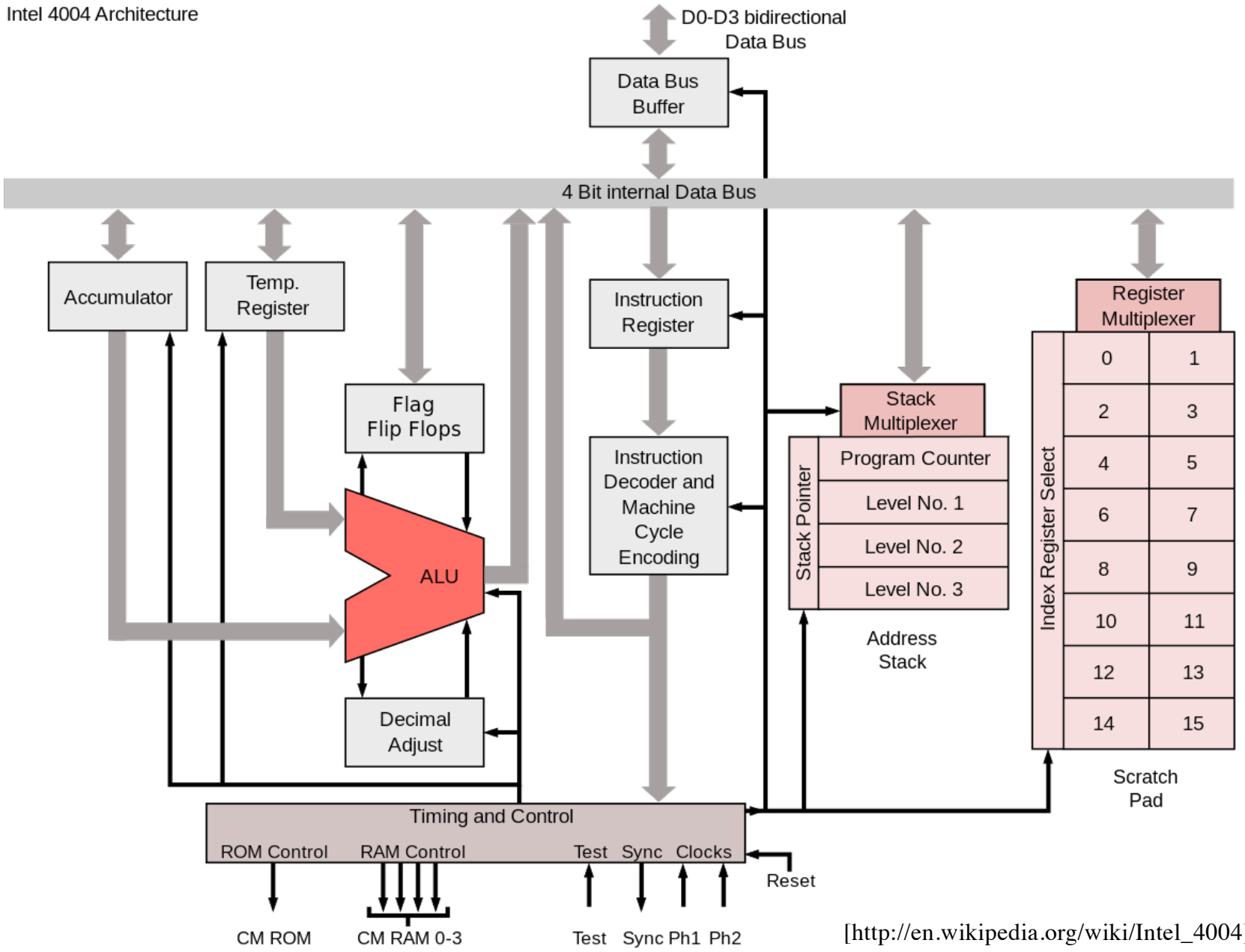
- **Maximum clock speed was 740 kHz**
- **Instruction cycle time: 10.8 μs**
  (8 clock cycles / instruction cycle)
- **Instruction execution time 1 or 2 instruction cycles (10.8 or 21.6 μs), 46300 to 92600 instructions per second**
- **Built using 2,300 transistors**

[http://en.wikipedia.org/wiki/Intel_4004]

Technical specifications

- **Separate program and data storage.**
- **The 4004, with its need to keep pin count down, used a single multiplexed 4-bit bus for transferring:**
  - **12-bit addresses**
  - **8-bit instructions**
  - **4-bit data words**
- **Instruction set contained 46 instructions (of which 41 were 8 bits wide and 5 were 16 bits wide)**
- **Register set contained 16 registers of 4 bits each**
- **Internal subroutine stack, 3 levels deep.**

[http://en.wikipedia.org/wiki/Intel_4004]



[http://en.wikipedia.org/wiki/Intel_4004]

Intel 4004 registers

$^{1}_{1}$ $^{1}_{0}$ $^{0}_{9}$ $^{0}_{8}$ $^{0}_{7}$ $^{0}_{6}$ $^{0}_{5}$ $^{0}_{4}$ $^{0}_{3}$ $^{0}_{2}$ $^{0}_{1}$ $^{0}_{0}$ *(bit position)*

Main registers

		A	**A**ccumulator
	R0	R1	
	R2	R3	
	R4	R5	
	R6	R7	
	R8	R9	
	R10	R11	
	R12	R13	
	R14	R15	

Program counter

PC	**P**rogram **C**ounter

Push-down address call stack

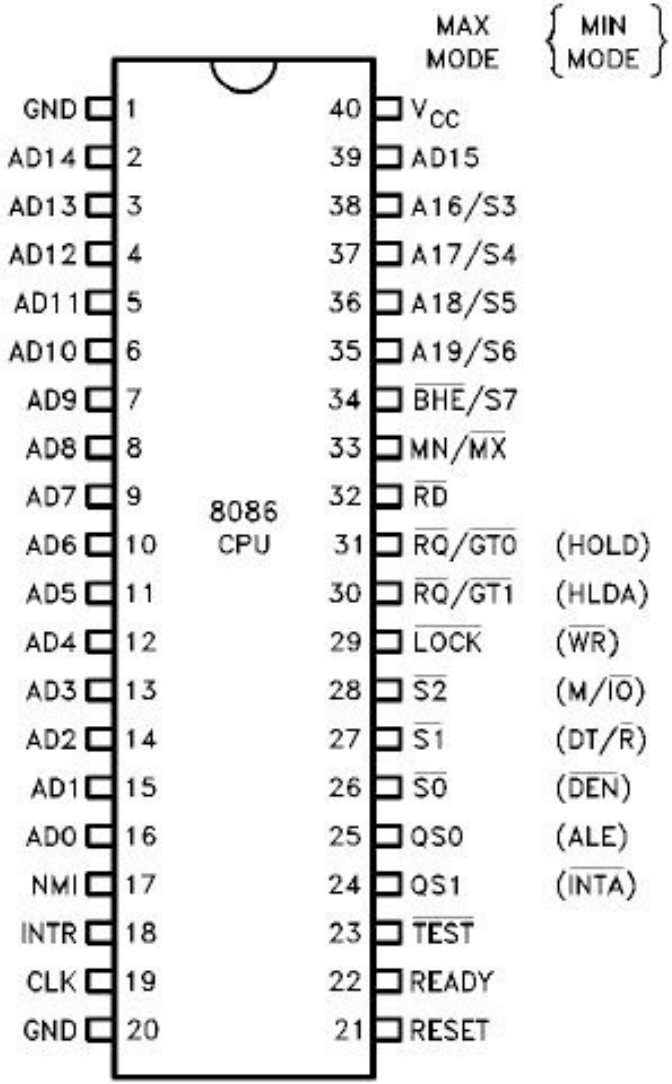
PC1	Call level 1
PC2	Call level 2
PC3	Call level 3

Status register

		C	P	Z	S	Flags

[http://en.wikipedia.org/wiki/Intel_4004]

Intel's 8086 Chip

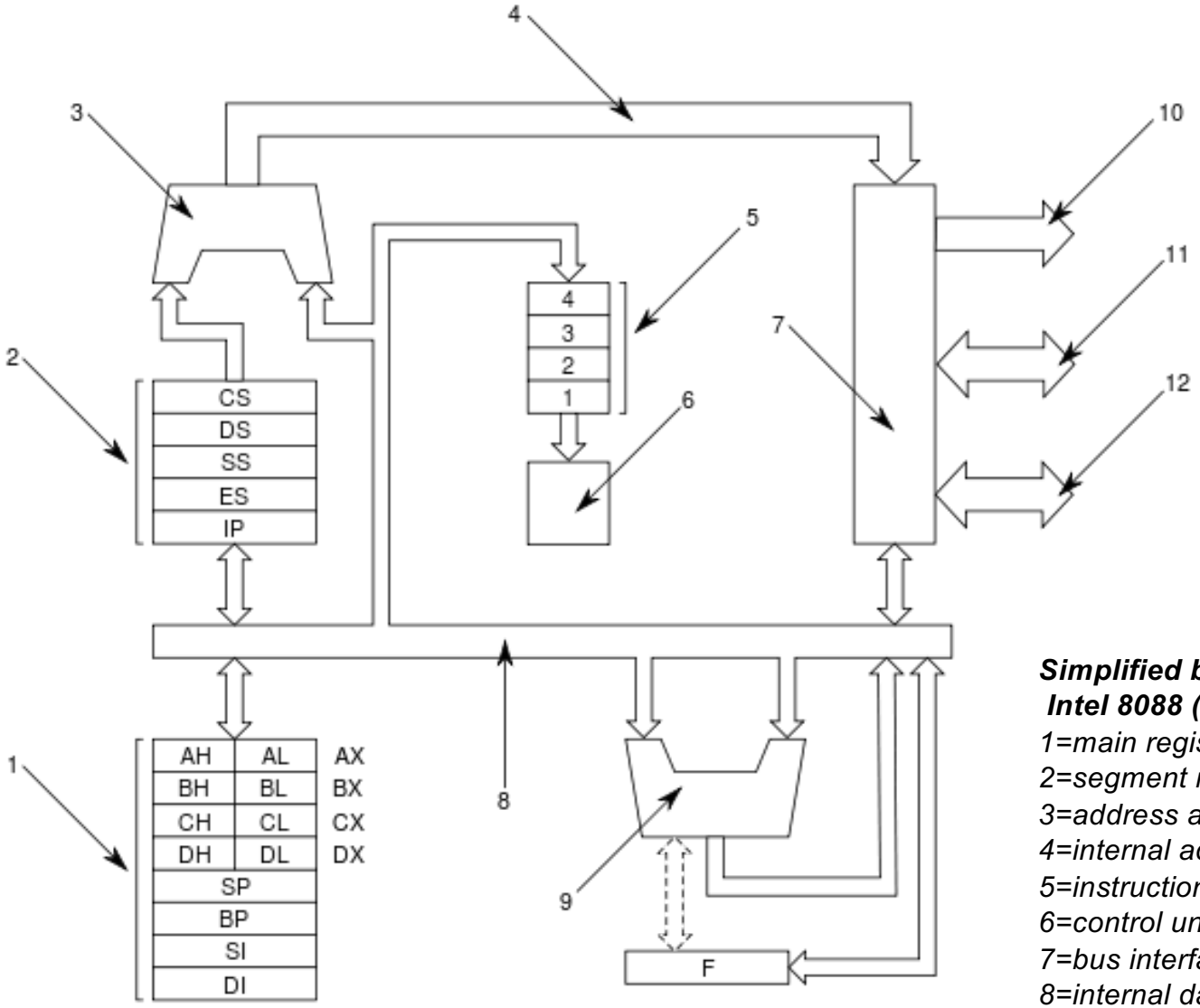


[http://en.wikipedia.org/wiki/Intel_8086]

Intel 8086 registers

$^{1}_{9}$ $^{1}_{8}$ $^{1}_{7}$ $^{1}_{6}$ $^{1}_{5}$ $^{1}_{4}$ $^{1}_{3}$ $^{1}_{2}$ $^{1}_{1}$ $^{1}_{0}$ $^{0}_{9}$ $^{0}_{8}$ $^{0}_{7}$ $^{0}_{6}$ $^{0}_{5}$ $^{0}_{4}$ $^{0}_{3}$ $^{0}_{2}$ $^{0}_{1}$ $^{0}_{0}$			*(bit position)*
Main registers			
	AH	AL	**AX** (primary accumulator)
	BH	BL	**BX** (base, accumulator)
	CH	CL	**CX** (counter, accumulator)
	DH	DL	**DX** (accumulator, other functions)
Index registers			
0 0 0 0	SI		**S**ource **I**ndex
0 0 0 0	DI		**D**estination **I**ndex
0 0 0 0	BP		**B**ase **P**ointer
0 0 0 0	SP		**S**tack **P**ointer
Program counter			
0 0 0 0	IP		**I**nstruction **P**ointer
Segment registers			
CS		0 0 0 0	**C**ode **S**egment
DS		0 0 0 0	**D**ata **S**egment
ES		0 0 0 0	**E**xtra**S**egment
SS		0 0 0 0	**S**tack **S**egment
Status register			
	- - - - O D I T S Z - A - P - C		Flags

[http://en.wikipedia.org/wiki/Intel_8086]



Simplified block diagram of Intel 8088 (a variant of 8086);
1=main registers;
2=segment registers and IP;
3=address adder;
4=internal address bus;
5=instruction queue;
6=control unit (very simplified!);
7=bus interface;
8=internal databus;
9=ALU;
10/11/12=external address/ data/control bus.

[http://en.wikipedia.org/wiki/Intel_8086]

Questions?

THE END