> 出于简化考虑，这里`zero`的值默认设置为1. 事实上，`zero`指示了`aluout`的结果，若为0，则`zero`的值为1，反之则为0.

> aluc 4bit 可以支持16种操作，怎么分配我们可以自己定义。为了一致性，我们定义部分指令的控制码字aluc为inst30+func3，u型就是特殊的，我们的定义只要不与其他指令冲突即可。

U型指令

`lui x10, 0`，`0x00000537`

inst[0] 1

zero 1

wmem 0

wreg 1

m2reg 0

aluc 0010

aluimm 1

pcsource 00

sext 0

i_lui 1

i_sw 0

shift 0

`11010001|01000100`

UJ型指令

`Call: jal sum`，`0x054000ef`

inst[0] 1

zero 1

wmem 0

wreg 1

m2reg 0

aluc xxxx

aluimm 0

pcsource 11

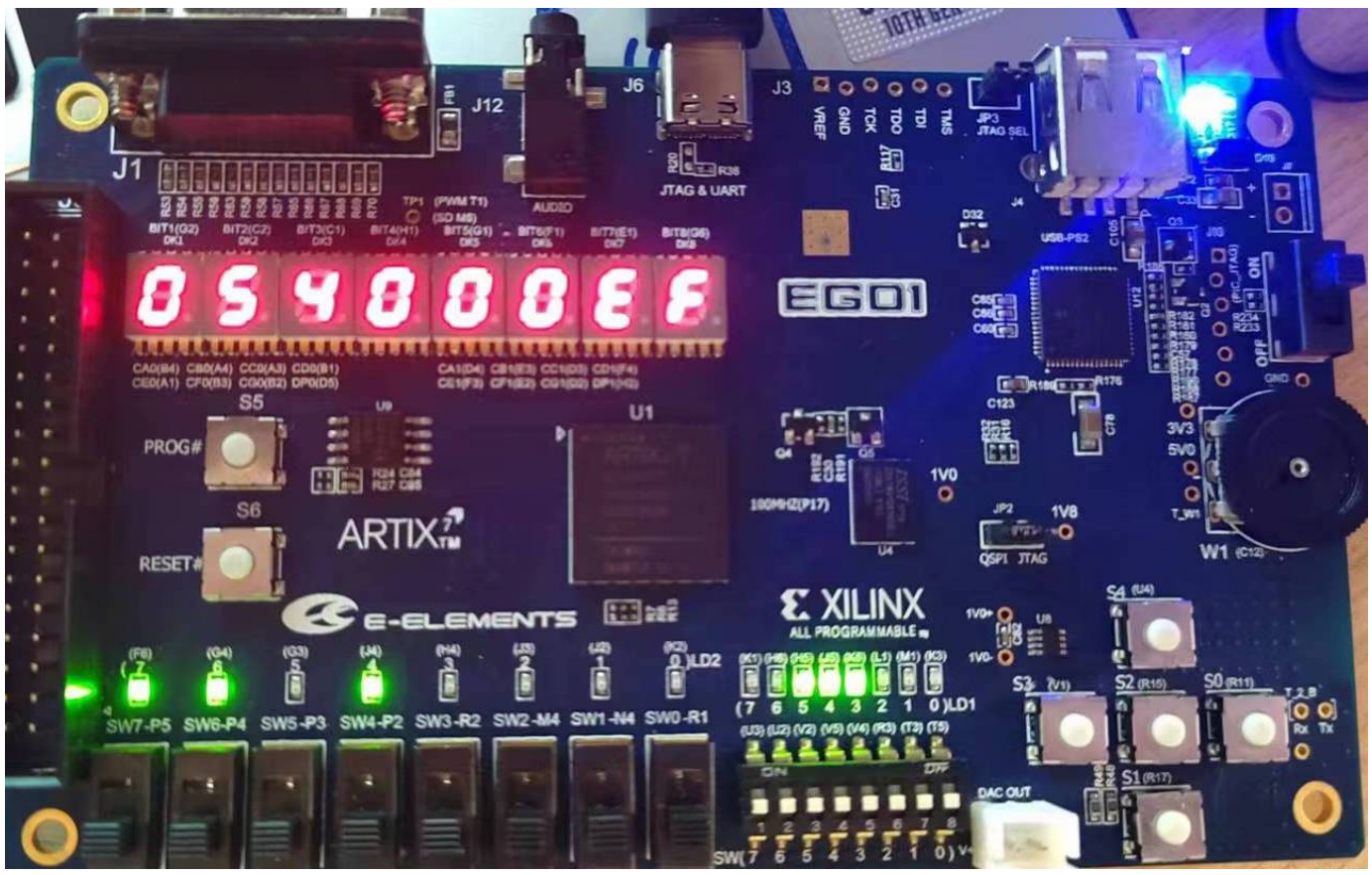
sext 1

i_lui 0

i_sw 0

shift 0

`11010xxxx|0111000`



R型指令

`xor x18, x20, x19`，`0x013a4933`

inst[0] 1

zero 1

wmem 0

wreg 1

m2reg 0

aluc 0100

aluimm 0

pcsource 00

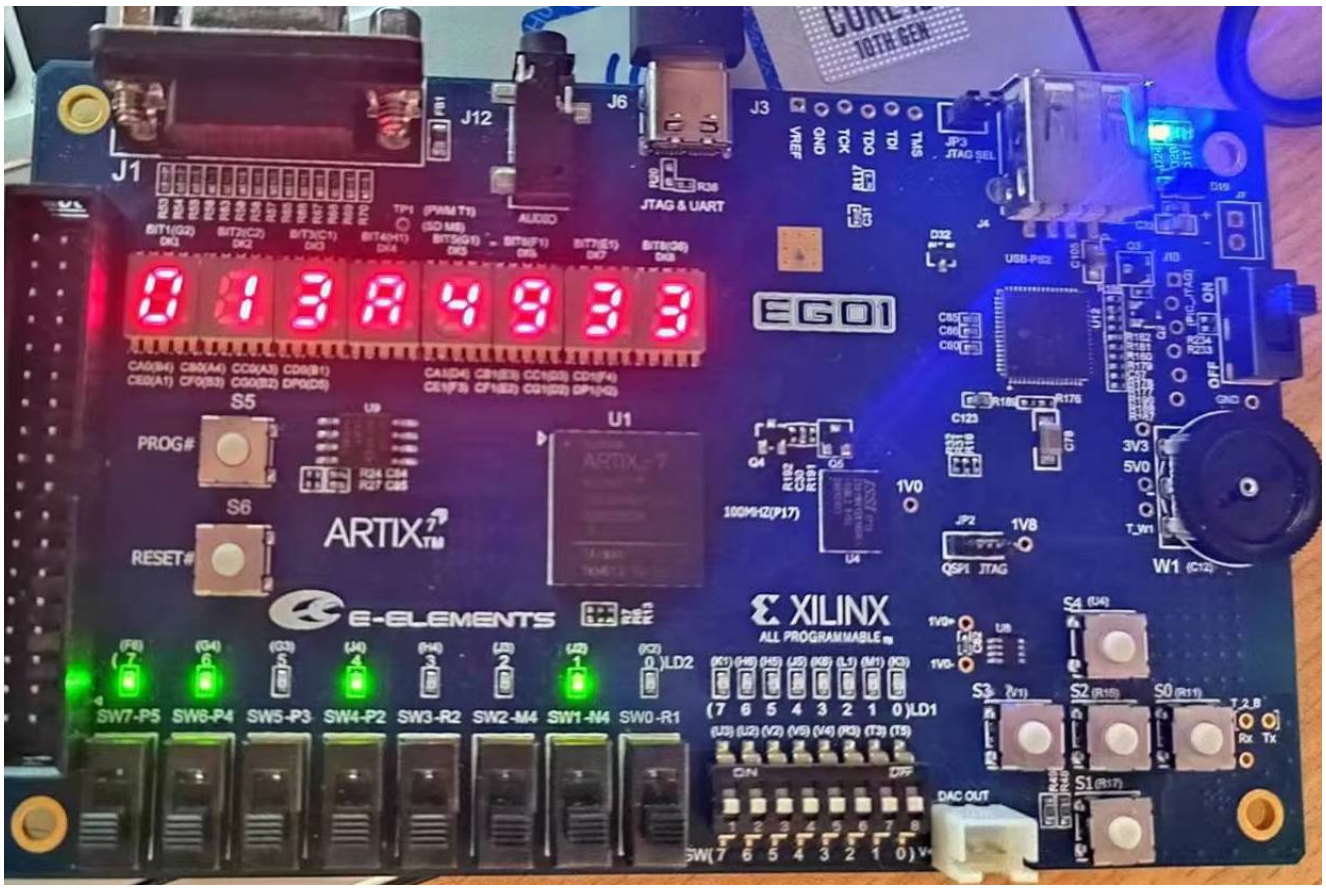
sext 0

i_lui 0

i_sw 0

shift 0

11010010|00000000



S型指令

`sw x12, 0(x4)`，`0x00C22023`

inst[0] 1

zero 1

wmem 1

wreg 0

m2reg 0

aluc 0000

aluimm 1

`pcsource` 00

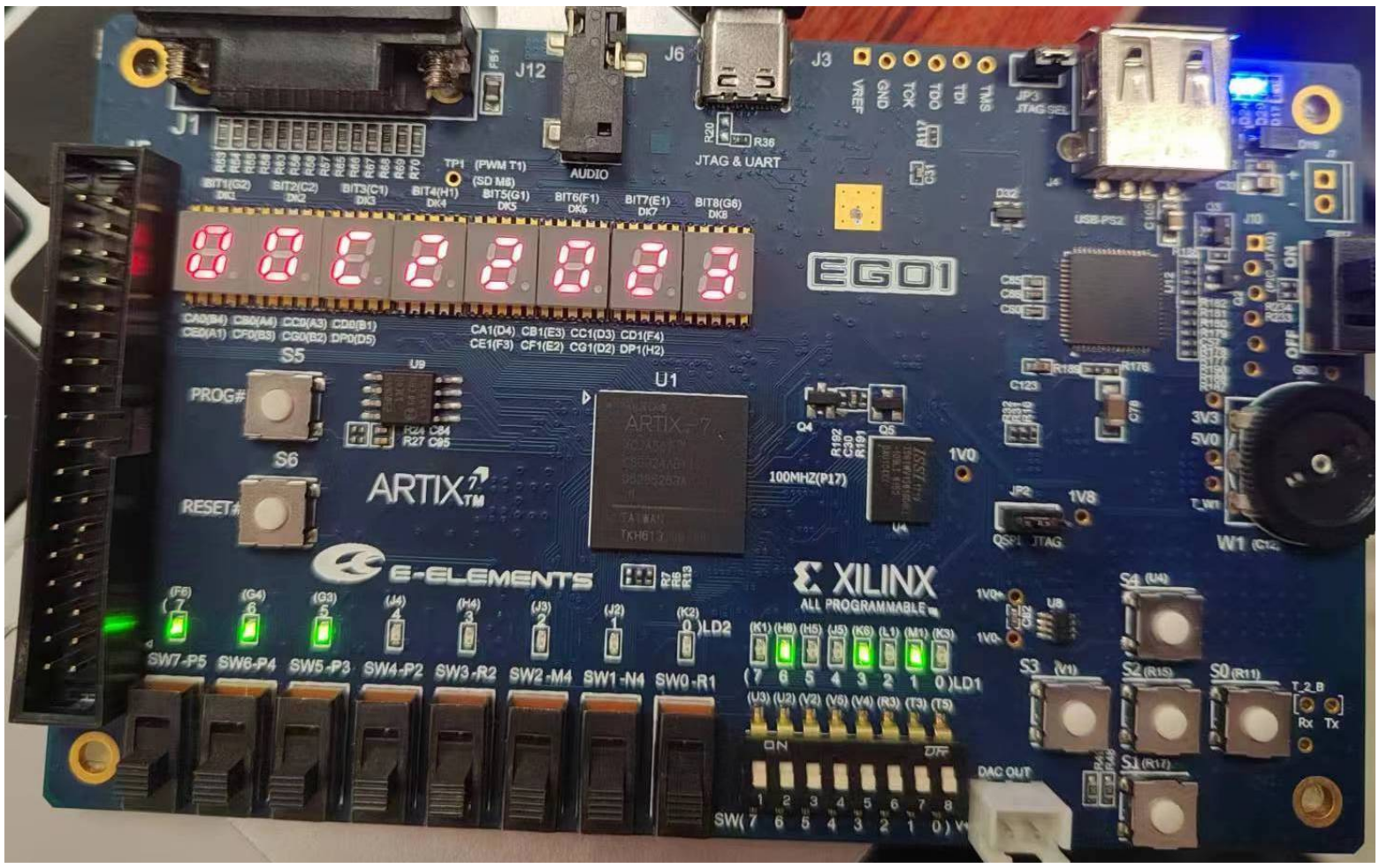
`sext` 1

`i_lui` 0

`i_sw` 1

`shift` 0

`11100000|01001010`



SB型指令

`beq x5, x0, shift` , `0x00028463`

`inst[0]` 1

`zero` 0 | 1

`wmem` 0

`wreg` 0

`m2reg` 0

`aluc` 1000

`aluimm` 0

`pcsource` 00 | 01

`sext` 1

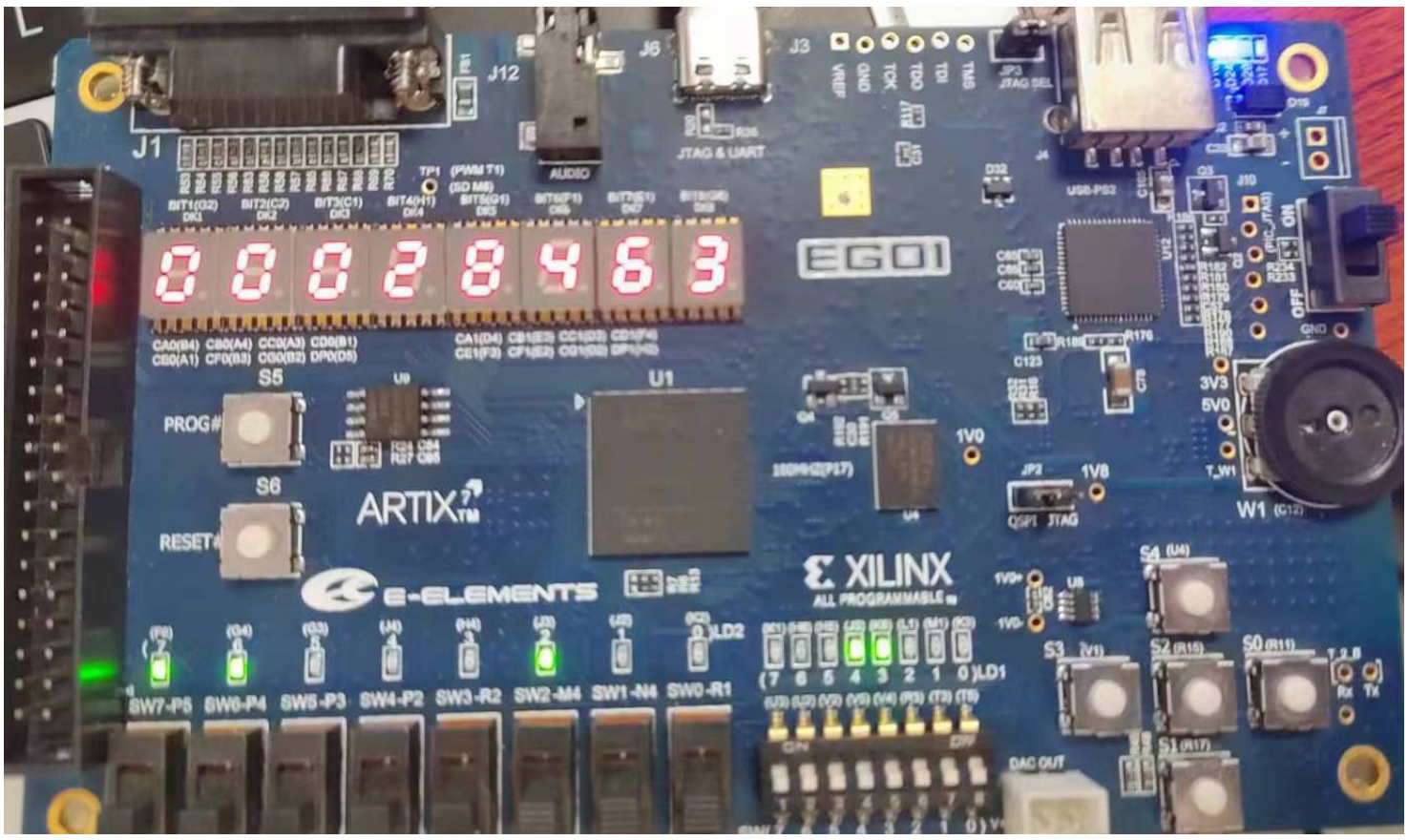
`i_lui` 0

`i_sw` 0

`shift` 0

不跳转：`10000100|00001000`

跳转：`11000100|00011000`， 此处是跳转的情况！



I型指令

`addi x25, x0, 1`，`0x00100C93`

`inst[0]` 1

`zero` 1

`wmem` 0

`wreg` 1

`m2reg` 0

`aluc` 0000

`aluimm` 1

`pcsource` 00

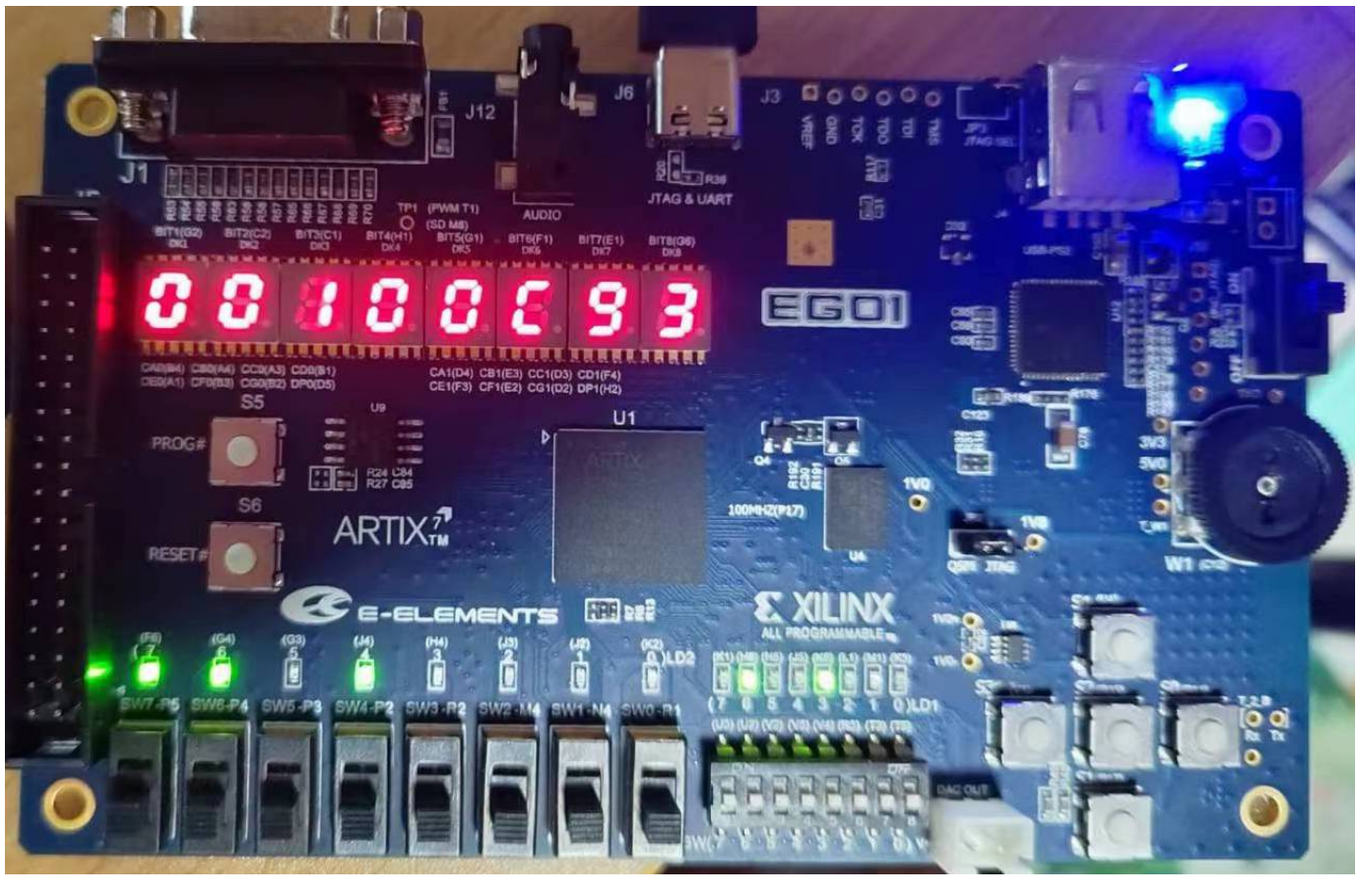
`sext` 1

`i_lui` 0

`i_sw` 0

`shift` 0

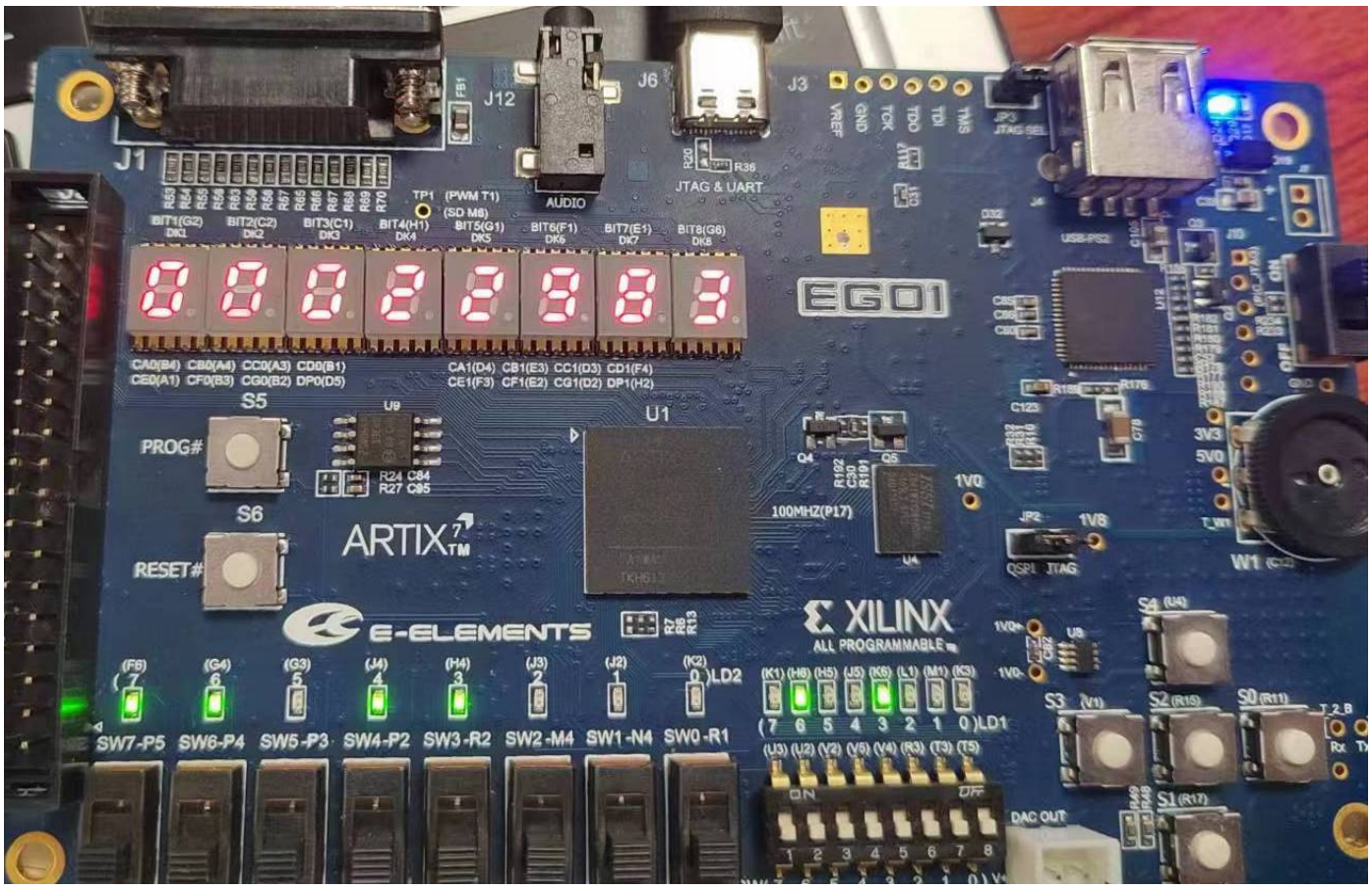
`11010000|01001000`



`lw s3, 0(tp)` , `0x00022983`

inst[0]	1
zero	1
wmem	0
wreg	1
m2reg	1
aluc	0000
aluimm	1
pcsource	00
sext	1
i_lui	0
i_sw	0
shift	0

`11010000|01001000`


J1
J12
J6
J3
AUDIO
JTAG & UART
EGO1
ARTIX7
E-ELEMENTS
XILINX
ALL PROGRAMMABLE
PROG#
RESET#
S5
S6
U1
100MHZ(P17)
SW7-P5
SW6-P4
SW5-P3
SW4-P2
SW3-R2
SW2-M4
SW1-N4
SW0-R1
DAC OUT