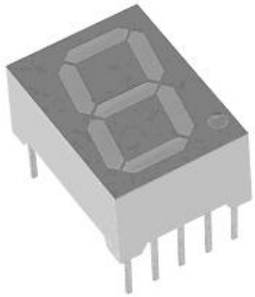
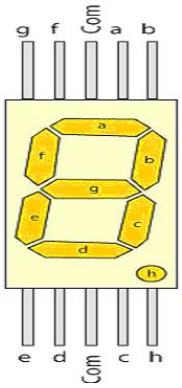
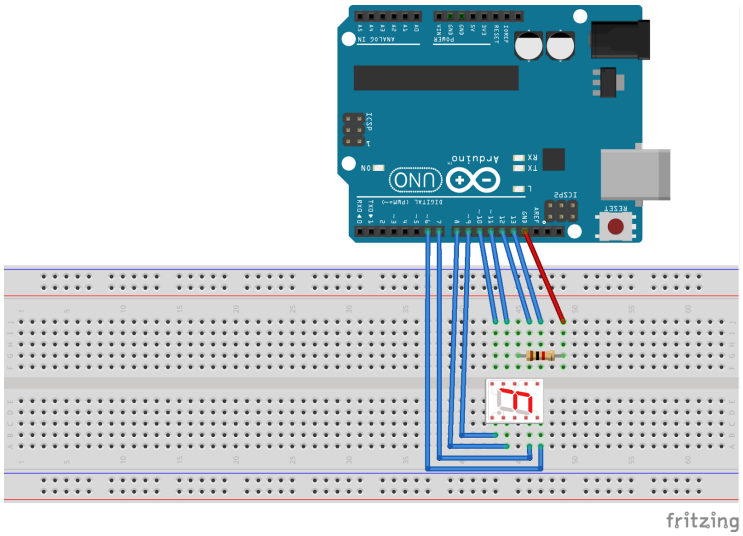
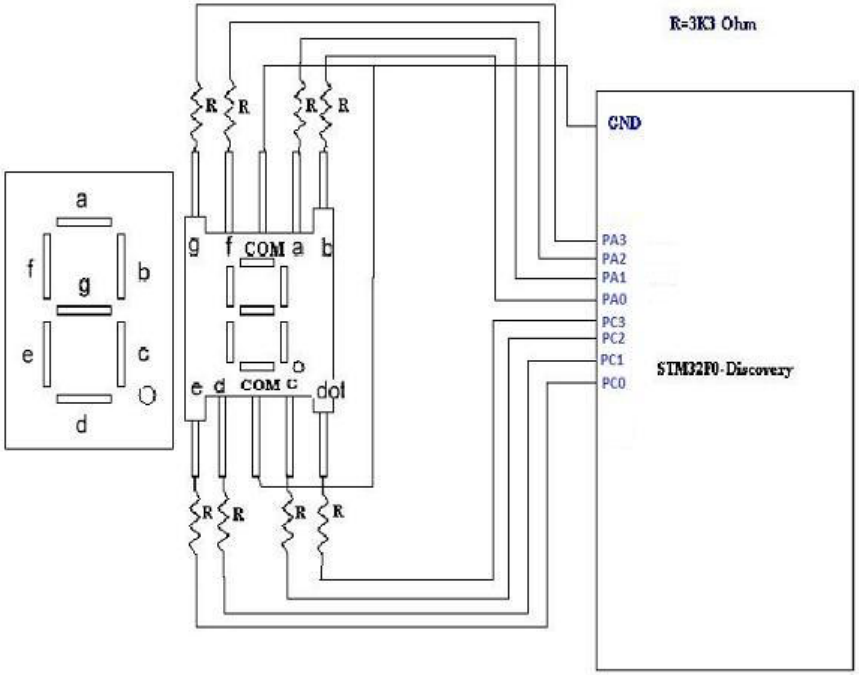
ARM GPIO 7-Segment

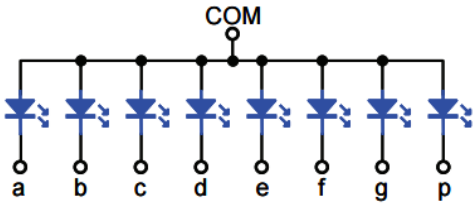
7-Segment



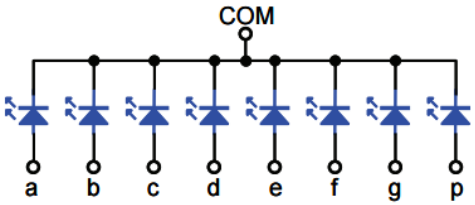








(a)共陽極結構

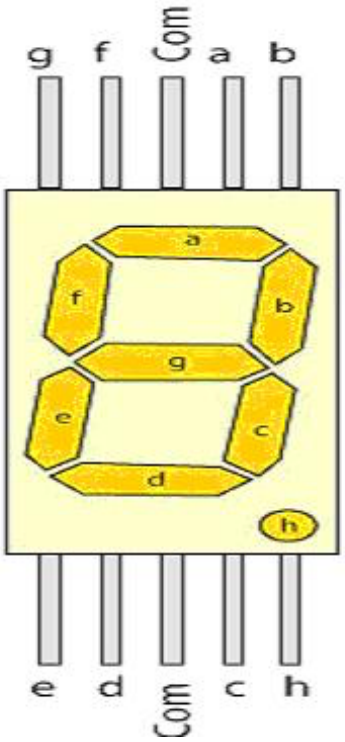


(b)共陰極結構

Display coding

PA3 g PA2 f PA1 a PA0 b

PC3 h PC2 c PC1 d PC0 e



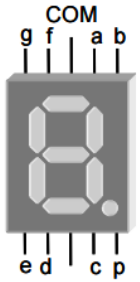
g	f	e	d	dp	c	b	a	number
0	1	1	1	0	1	1	1	0
0	0	0	0	0	1	1	0	1
1	0	1	1	0	0	1	1	2
1	0	0	1	0	1	1	1	3
0	1	1	0	0	1	1	0	4
1	1	0	1	0	1	0	1	5
1	1	1	1	0	1	0	1	6
0	0	0	0	0	1	1	1	7
1	1	1	1	0	1	1	1	8
1	1	0	1	0	1	1	1	9

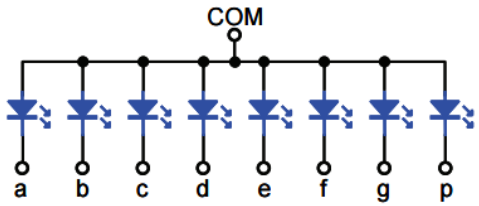
g	f	e	d	dp	c	b	a	number	hex code PORT A	&	PORT C
0	1	1	1	0	1	1	1	0	0X0007	&	0X0007
0	0	0	0	0	1	1	0	1	0X0001	&	0X0004
1	0	1	1	0	0	1	1	2	0X000B	&	0X0003
1	0	0	1	0	1	1	1	3	0X000B	&	0X0006
0	1	1	0	0	1	1	0	4	0X000D	&	0X0004
1	1	0	1	0	1	0	1	5	0X000E	&	0X0006
1	1	1	1	0	1	0	1	6	0X000E	&	0X0007
0	0	0	0	0	1	1	1	7	0X0003	&	0X0004
1	1	1	1	0	1	1	1	8	0X000F	&	0X0007
1	1	0	1	0	1	1	1	9	0X000F	&	0X0006

```
int main(void)
{
Init_GPIO();
while(1)
{
GPIOA->ODR = 0x0007;
GPIOC->ODR = 0x0007;
delay_ms(1000);

GPIOA->ODR = 0x0001;
GPIOC->ODR = 0x0004;
delay_ms(1000);

GPIOA->ODR = 0x000B;
GPIOC->ODR = 0x0003;
delay_ms(1000);

…
}
}
```

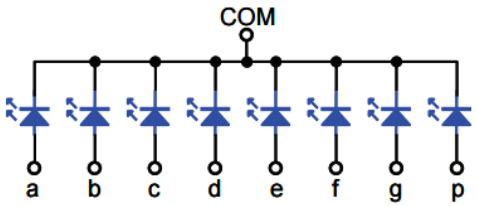
```
void Init_GPIO(void)
{ GPIO_InitTypeDef GPIO_InitStructure;
RCC_AHBPeriphClockCmd(RCC_AHBPeriph_GPIOA |
RCC_AHBPeriph_GPIOC, ENABLE);

GPIO_InitStructure.GPIO_Pin =GPIO_Pin_0 |
GPIO_Pin_1 | GPIO_Pin_2 | GPIO_Pin_3 ;
GPIO_InitStructure.GPIO_Speed =
GPIO_Speed_10MHz;
GPIO_InitStructure.GPIO_Mode =
GPIO_Mode_OUT;
GPIO_InitStructure.GPIO_OType =
GPIO_OType_PP;
GPIO_Init(GPIOC, &GPIO_InitStructure);

GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0 |
GPIO_Pin_1 | GPIO_Pin_2 | GPIO_Pin_3 ;
GPIO_InitStructure.GPIO_Speed =
GPIO_Speed_10MHz;
GPIO_InitStructure.GPIO_Mode =
GPIO_Mode_OUT;
GPIO_InitStructure.GPIO_OType =
GPIO_OType_PP;
GPIO_Init(GPIOA, &GPIO_InitStructure);}
```

7-Seg LED

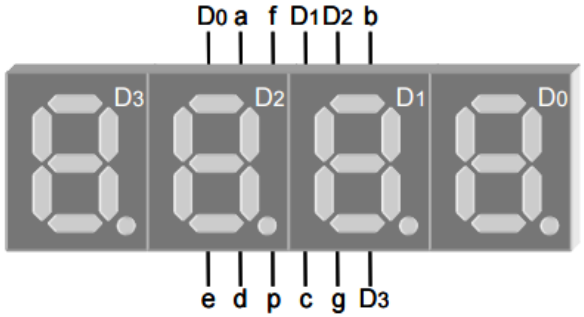


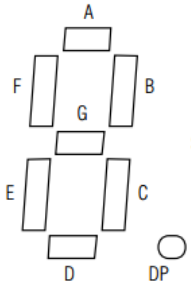


(a)共陽極結構



(b)共陰極結構





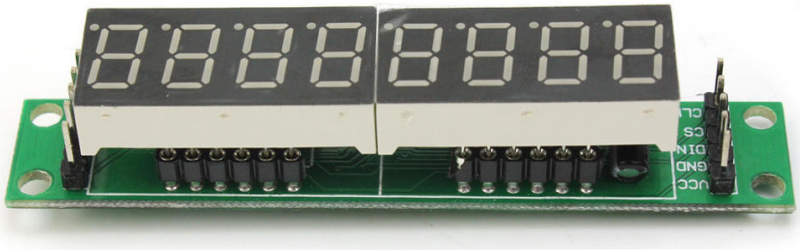
STANDARD 7-SEGMENT LED

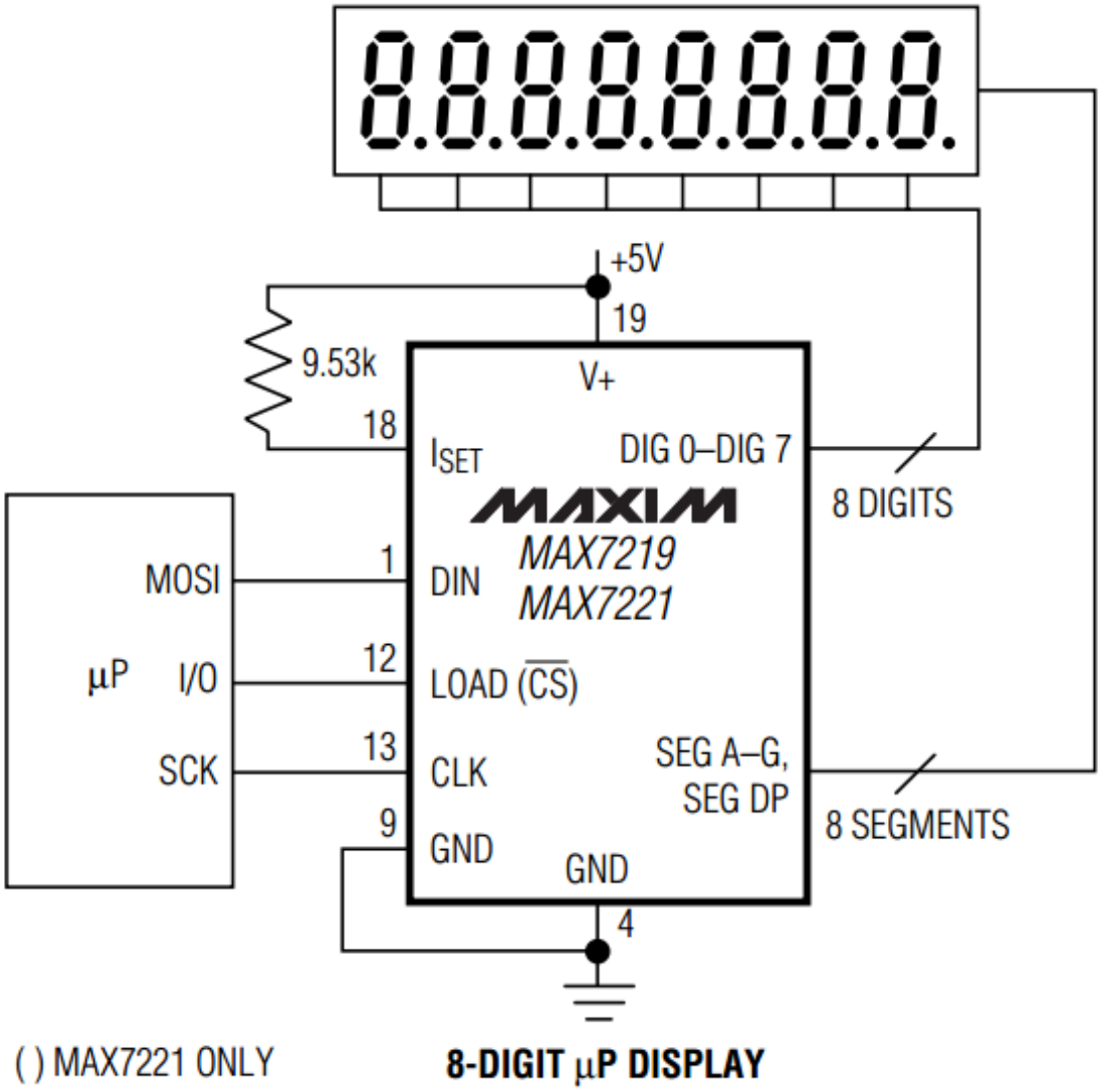
If we connect stm32 I/O pin on 7-Seg LED directly

- We use eight 7-Seg LED ➔ We will need 16 GPIO pin!
- We have to scan eight 7-Seg LED to show different number on it!

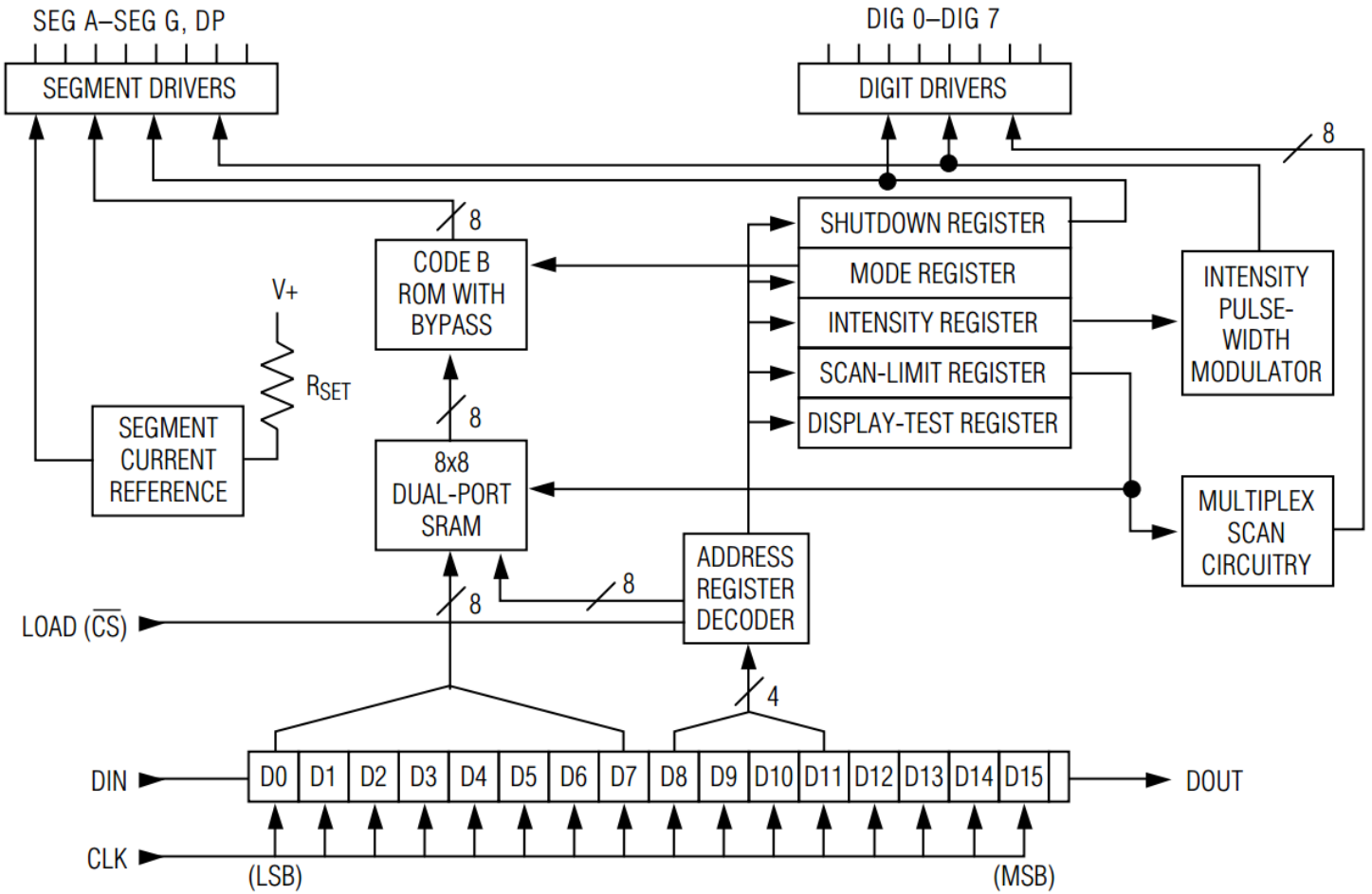
➔We use Max7219 to simplify our work!!

Max7219




+5V
19
9.53k
V+
18
ISET
DIG 0–DIG 7
8 DIGITS
MAXIM
MAX7219
MAX7221
MOSI
1
DIN
µP
I/O
12
LOAD (CS)
SCK
13
CLK
SEG A–G,
SEG DP
8 SEGMENTS
9
GND
GND
4
( ) MAX7221 ONLY
8-DIGIT µP DISPLAY

Max7219


SEG A–SEG G, DP
SEGMENT DRIVERS
DIG 0–DIG 7
DIGIT DRIVERS
8
8
CODE B ROM WITH BYPASS
SHUTDOWN REGISTER
MODE REGISTER
INTENSITY REGISTER
SCAN-LIMIT REGISTER
DISPLAY-TEST REGISTER
INTENSITY PULSE-WIDTH MODULATOR
V+
RSET
SEGMENT CURRENT REFERENCE
8
8x8 DUAL-PORT SRAM
MULTIPLEX SCAN CIRCUITRY
8
8
ADDRESS REGISTER DECODER
LOAD (CS)
4
DIN
D0
D1
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
D13
D14
D15
DOUT
CLK
(LSB)
(MSB)

Max7219

- DIN: Serial-Data Input. Data is loaded into the internal 16-bit shift register on CLK's rising edge.
- CS: Load-Data Input. The last 16 bits of serial data are latched on LOAD(CS)'s rising edge.
- CLK: Serial-Clock Input. 10MHz maximum rate. On CLK's rising edge, data is shifted into the internal shift register.

Table 1. Serial-Data Format (16 Bits)

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	ADDRESS				MSB	DATA						LSB

Max7219

Table 2. Register Address Map

REGISTER	ADDRESS					HEX CODE
	D15–D12	D11	D10	D9	D8	
No-Op	X	0	0	0	0	0xX0
Digit 0	X	0	0	0	1	0xX1
Digit 1	X	0	0	1	0	0xX2
Digit 2	X	0	0	1	1	0xX3
Digit 3	X	0	1	0	0	0xX4
Digit 4	X	0	1	0	1	0xX5
Digit 5	X	0	1	1	0	0xX6
Digit 6	X	0	1	1	1	0xX7
Digit 7	X	1	0	0	0	0xX8
Decode Mode	X	1	0	0	1	0xX9
Intensity	X	1	0	1	0	0xXA
Scan Limit	X	1	0	1	1	0xXB
Shutdown	X	1	1	0	0	0xXC
Display Test	X	1	1	1	1	0xXF

Max7219—Shutdown Register

Table 3. Shutdown Register Format (Address (Hex) = 0xXC)

MODE	ADDRESS CODE (HEX)	REGISTER DATA							
		D7	D6	D5	D4	D3	D2	D1	D0
Shutdown Mode	0xXC	X	X	X	X	X	X	X	0
Normal Operation	0xXC	X	X	X	X	X	X	X	1

When the MAX7219 is in shutdown mode, the scan oscillator is halted, all segment current sources are pulled to ground, and all digit drivers are pulled to V+, thereby blanking the display. Data in the digit and control registers remains unaltered.

Max7219—Decode-Mode Register

Table 4. Decode-Mode Register Examples (Address (Hex) = 0xX9)

DECODE MODE	REGISTER DATA								HEX CODE
	D7	D6	D5	D4	D3	D2	D1	D0	
No decode for digits 7–0	0	0	0	0	0	0	0	0	0x00
Code B decode for digit 0 No decode for digits 7–1	0	0	0	0	0	0	0	1	0x01
Code B decode for digits 3–0 No decode for digits 7–4	0	0	0	0	1	1	1	1	0x0F
Code B decode for digits 7–0	1	1	1	1	1	1	1	1	0xFF

Max7219—Decode-Mode Register

Table 5. Code B Font

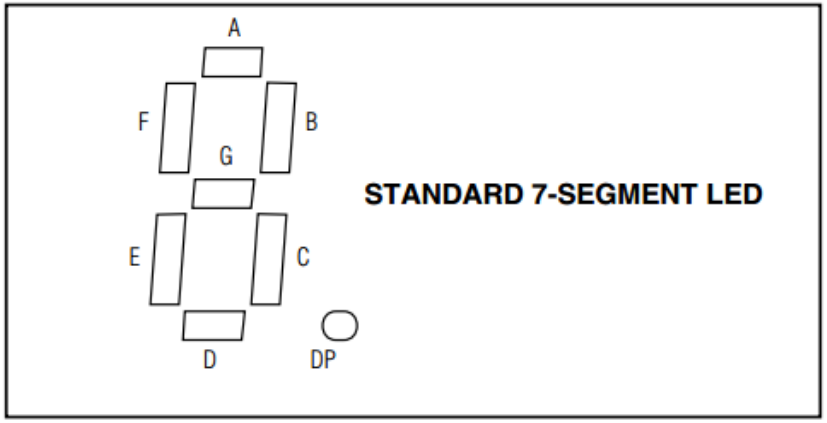
7-SEGMENT CHARACTER	REGISTER DATA						ON SEGMENTS = 1							
	D7*	D6–D4	D3	D2	D1	D0	DP*	A	B	C	D	E	F	G
0		X	0	0	0	0		1	1	1	1	1	1	0
1		X	0	0	0	1		0	1	1	0	0	0	0
2		X	0	0	1	0		1	1	0	1	1	0	1
3		X	0	0	1	1		1	1	1	1	0	0	1
4		X	0	1	0	0		0	1	1	0	0	1	1
5		X	0	1	0	1		1	0	1	1	0	1	1
6		X	0	1	1	0		1	0	1	1	1	1	1
7		X	0	1	1	1		1	1	1	0	0	0	0
8		X	1	0	0	0		1	1	1	1	1	1	1
9		X	1	0	0	1		1	1	1	1	0	1	1
—		X	1	0	1	0		0	0	0	0	0	0	1
E		X	1	0	1	1		1	0	0	1	1	1	1
H		X	1	1	0	0		0	1	1	0	1	1	1
L		X	1	1	0	1		0	0	0	1	1	1	0
P		X	1	1	1	0		1	1	0	0	1	1	1
blank		X	1	1	1	1		0	0	0	0	0	0	0

*The decimal point is set by bit D7 = 1

When the code B decode mode is used, the decoder looks only at the lower nibble of the data in the digit registers (D3–D0), disregarding bits D4–D6. D7, which sets the decimal point (SEG DP), is independent of the decoder and is positive logic (D7 = 1 turns the decimal point on)

Max7219—Decode-Mode Register

Table 6. No-Decode Mode Data Bits and Corresponding Segment Lines



	REGISTER DATA							
	D7	D6	D5	D4	D3	D2	D1	D0
Corresponding Segment Line	DP	A	B	C	D	E	F	G

When no-decode is selected, data bits D7–D0 correspond to the segment lines of the MAX7219/MAX7221.

Max7219—Intensity Register

Table 7. Intensity Register Format (Address (Hex) = 0xXA)

DUTY CYCLE		D7	D6	D5	D4	D3	D2	D1	D0	HEX CODE
MAX7219	MAX7221									
1/32 (min on)	1/16 (min on)	X	X	X	X	0	0	0	0	0xX0
3/32	2/16	X	X	X	X	0	0	0	1	0xX1
5/32	3/16	X	X	X	X	0	0	1	0	0xX2
7/32	4/16	X	X	X	X	0	0	1	1	0xX3
9/32	5/16	X	X	X	X	0	1	0	0	0xX4
11/32	6/16	X	X	X	X	0	1	0	1	0xX5
13/32	7/16	X	X	X	X	0	1	1	0	0xX6
15/32	8/16	X	X	X	X	0	1	1	1	0xX7
17/32	9/16	X	X	X	X	1	0	0	0	0xX8
19/32	10/16	X	X	X	X	1	0	0	1	0xX9
21/32	11/16	X	X	X	X	1	0	1	0	0xXA
23/32	12/16	X	X	X	X	1	0	1	1	0xXB
25/32	13/16	X	X	X	X	1	1	0	0	0xXC
27/32	14/16	X	X	X	X	1	1	0	1	0xXD
29/32	15/16	X	X	X	X	1	1	1	0	0xXE
31/32	15/16 (max on)	X	X	X	X	1	1	1	1	0xXF

Max7219—Scan-Limit Register

Table 8. Scan-Limit Register Format (Address (Hex) = 0xXB)

SCAN LIMIT	REGISTER DATA								HEX CODE
	D7	D6	D5	D4	D3	D2	D1	D0	
Display digit 0 only*	X	X	X	X	X	0	0	0	0xX0
Display digits 0 & 1*	X	X	X	X	X	0	0	1	0xX1
Display digits 0 1 2*	X	X	X	X	X	0	1	0	0xX2
Display digits 0 1 2 3	X	X	X	X	X	0	1	1	0xX3
Display digits 0 1 2 3 4	X	X	X	X	X	1	0	0	0xX4
Display digits 0 1 2 3 4 5	X	X	X	X	X	1	0	1	0xX5
Display digits 0 1 2 3 4 5 6	X	X	X	X	X	1	1	0	0xX6
Display digits 0 1 2 3 4 5 6 7	X	X	X	X	X	1	1	1	0xX7

*See *Scan-Limit Register* section for application.

The scan-limit register sets how many digits are displayed, from 1 to 8. The number of scanned digits affects the display brightness,

Max7219—Display Test Register

Table 10. Display-Test Register Format (Address (Hex) = 0xXF)

MODE	REGISTER DATA							
	D7	D6	D5	D4	D3	D2	D1	D0
Normal Operation	X	X	X	X	X	X	X	0
Display Test Mode	X	X	X	X	X	X	X	1

Note: The MAX7219/MAX7221 remain in display-test mode (all LEDs on) until the display-test register is reconfigured for normal operation.

The display-test register operates in two modes: normal and display test. Display-test mode turns all LEDs on by overriding, but not altering, all controls and digit registers (including the shutdown register).

Max7219—register 功能整理

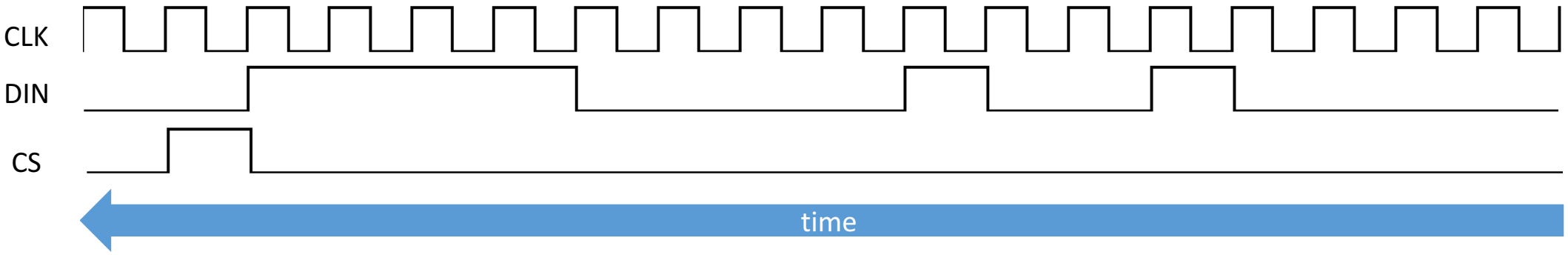
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	ADDRESS				MSB	DATA						LSB

- Decode Mode: 被設定為Decode Mode的數會透過解碼器分別將D0~D3的值解碼成7-Seg LED的0~9,-,E,H,L,P,(空白)，非Decode Mode的數則會將D0~D7直接顯示在7-Seg LED上(請參考Table6的圖)。
- Intensity: 用來設定7-Seg LED的亮度，0到15越來越亮。
- Scan Limit: 用來設定7-Seg LED的顯示位數0表示顯示一位，1表示顯示兩位，以此類推。
- Shutdown: 設為shutdown mode時7-Seg LED會關掉，是一種省電模式。
- Display Test:測試用！會讓所有LED都亮起來!!

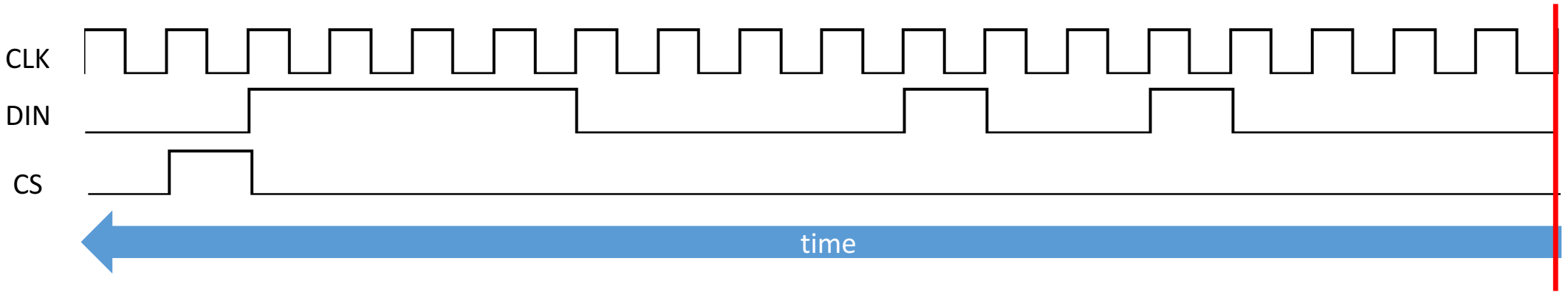
Example

I want to set decode mode(Code B decode for digit 0-3, no decode for digits 4-7), thus I have to set Serial-Data as below!
And then send a rising edge on CS pin to latch the Serial-Data!

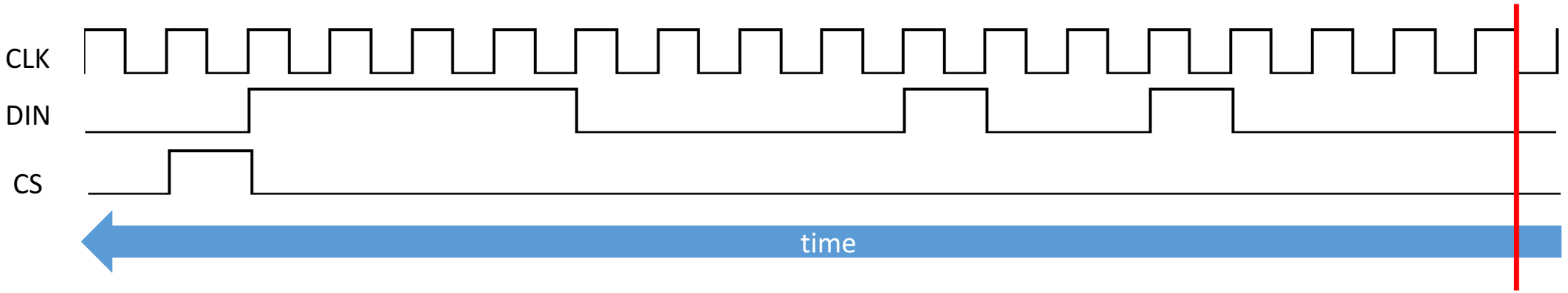
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	ADDRESS				DATA							
X	X	X	X	1	0	0	1	0	0	0	0	1	1	1	1



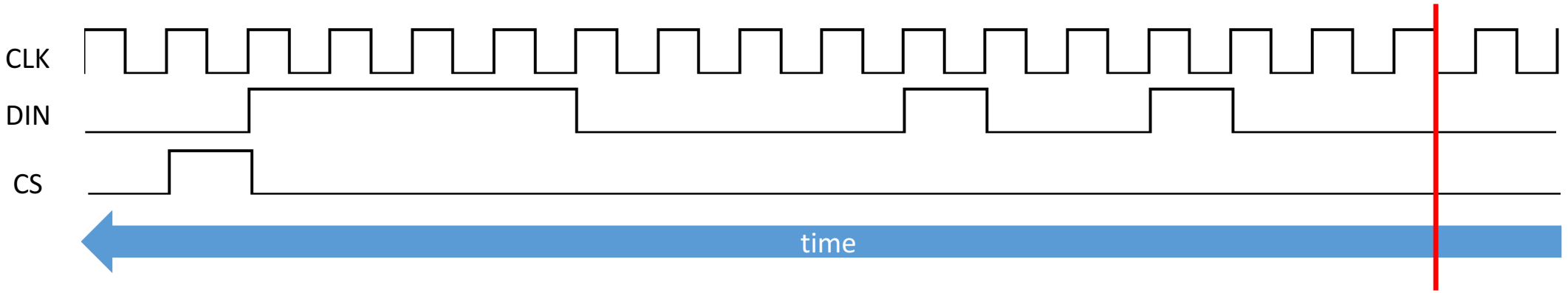
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	ADDRESS				DATA							
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X



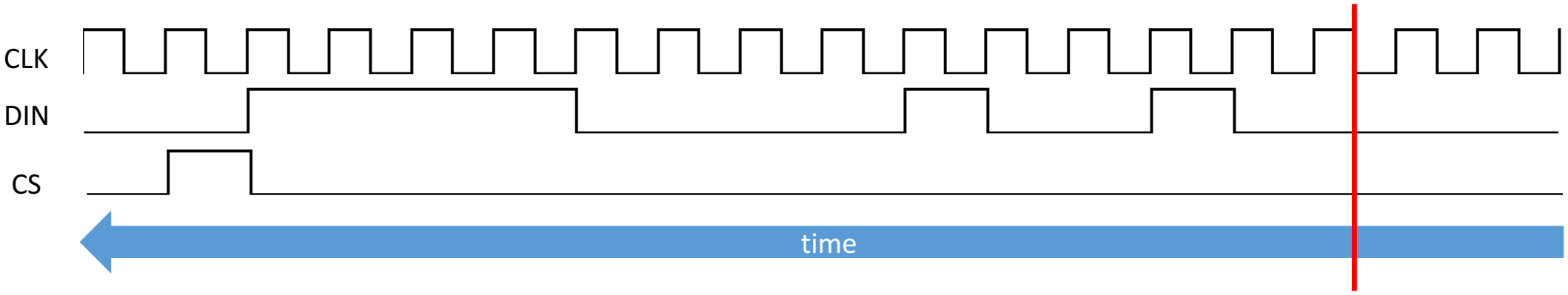
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	ADDRESS				DATA							
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0



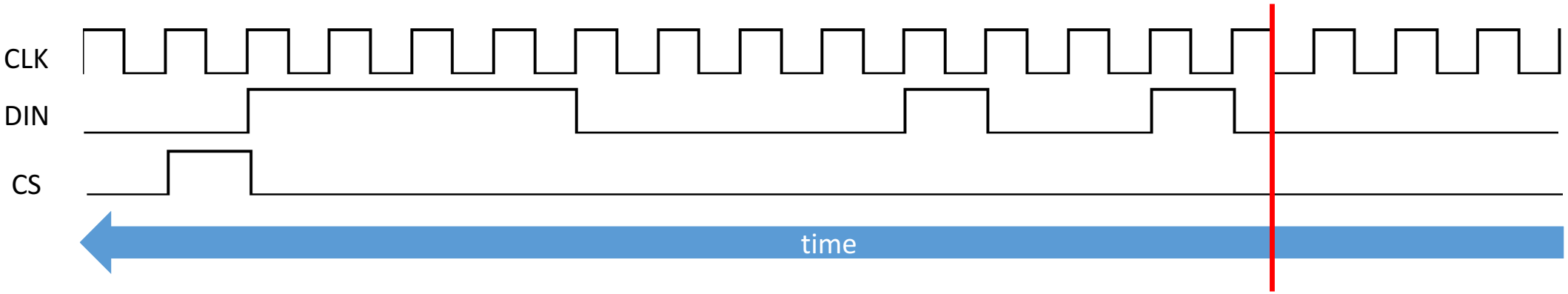
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	ADDRESS				DATA							
X	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0



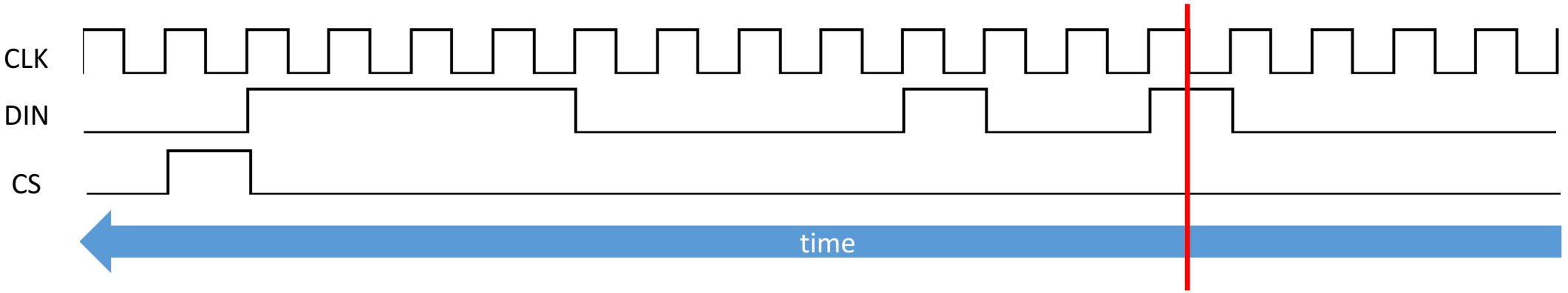
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	ADDRESS				DATA							
X	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0



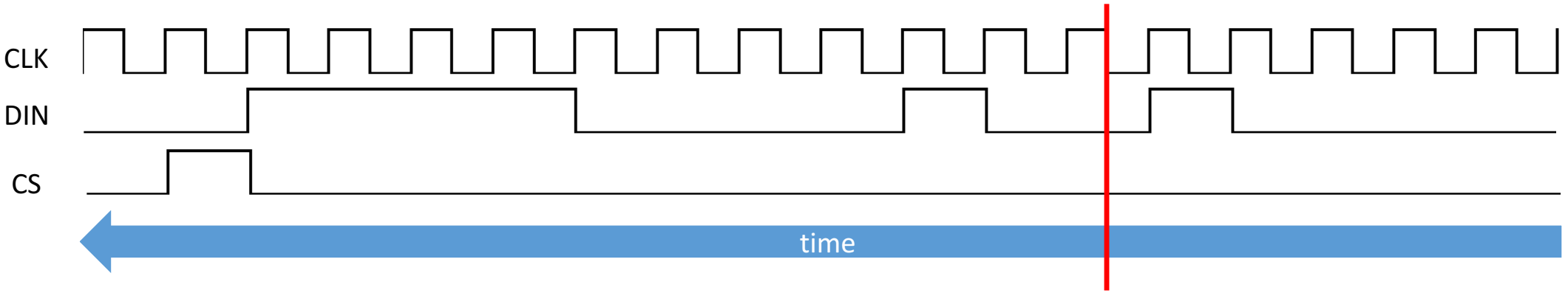
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	ADDRESS				DATA							
X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0



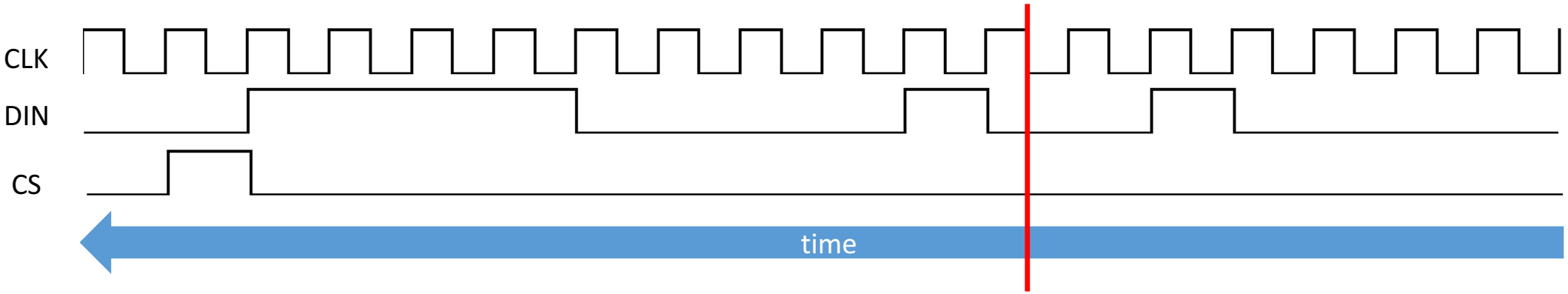
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	ADDRESS				DATA							
X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	1



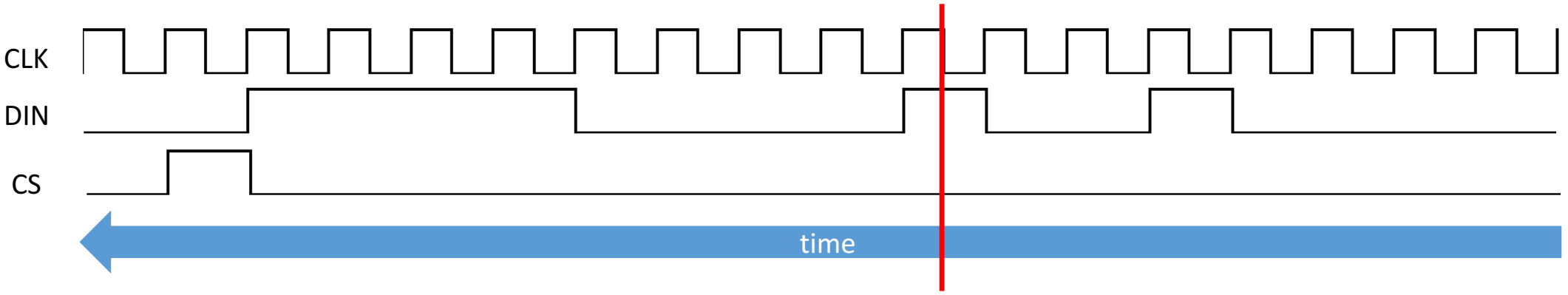
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	ADDRESS				DATA							
X	X	X	X	X	X	X	X	X	X	0	0	0	0	1	0



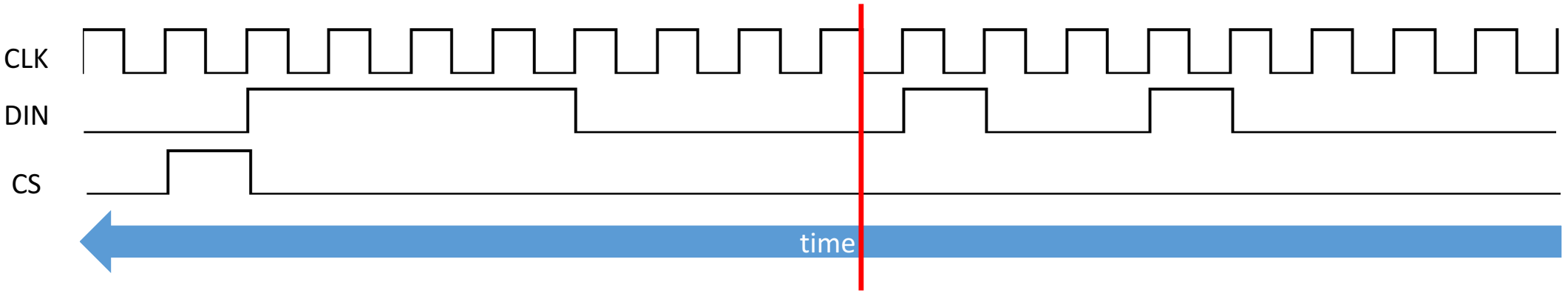
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	ADDRESS				DATA							
X	X	X	X	X	X	X	X	X	0	0	0	0	1	0	0



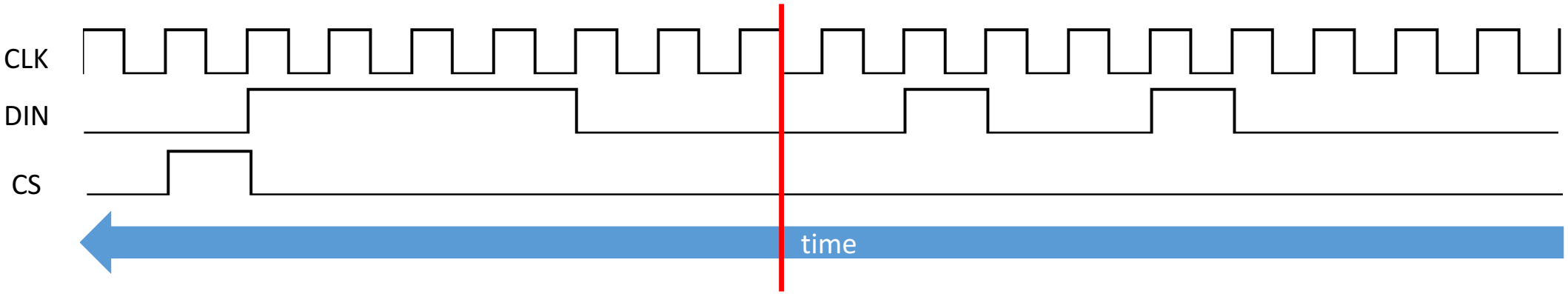
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	ADDRESS				DATA							
X	X	X	X	X	X	X	X	0	0	0	0	1	0	0	1



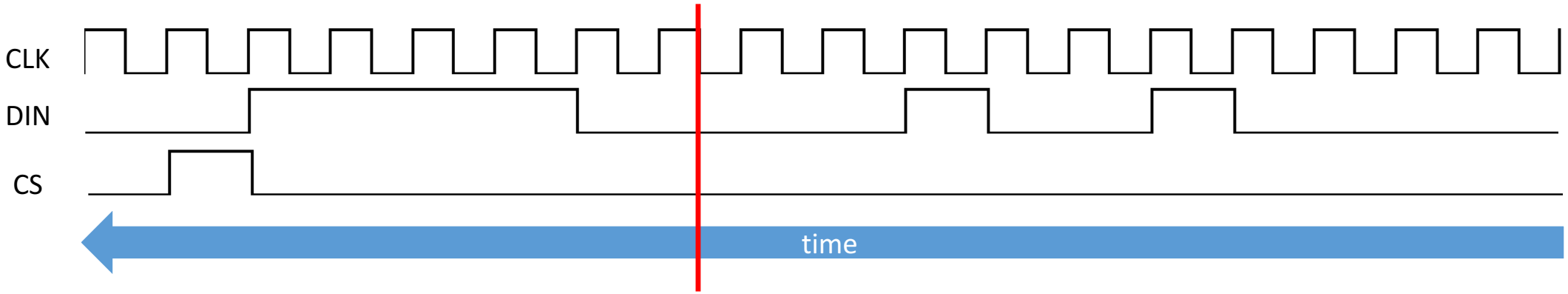
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	ADDRESS				DATA							
X	X	X	X	X	X	X	0	0	0	0	1	0	0	1	0



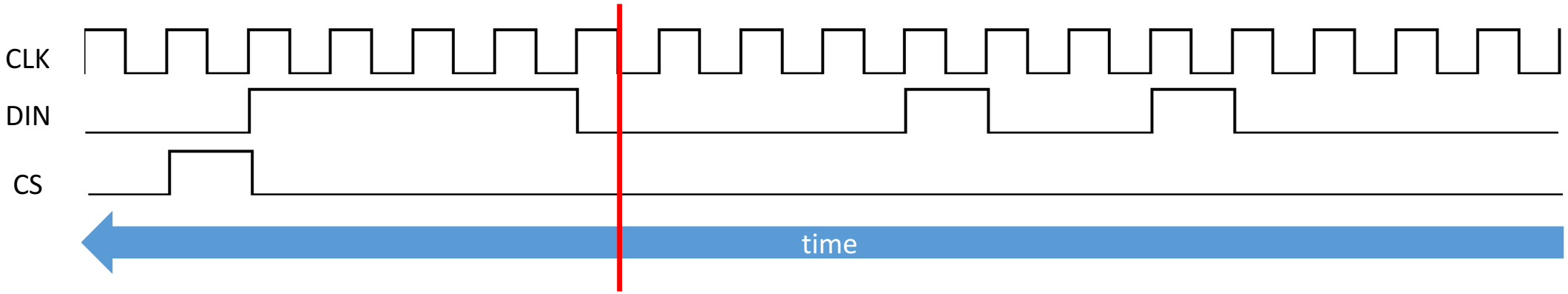
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	ADDRESS				DATA							
X	X	X	X	X	X	0	0	0	0	1	0	0	1	0	0



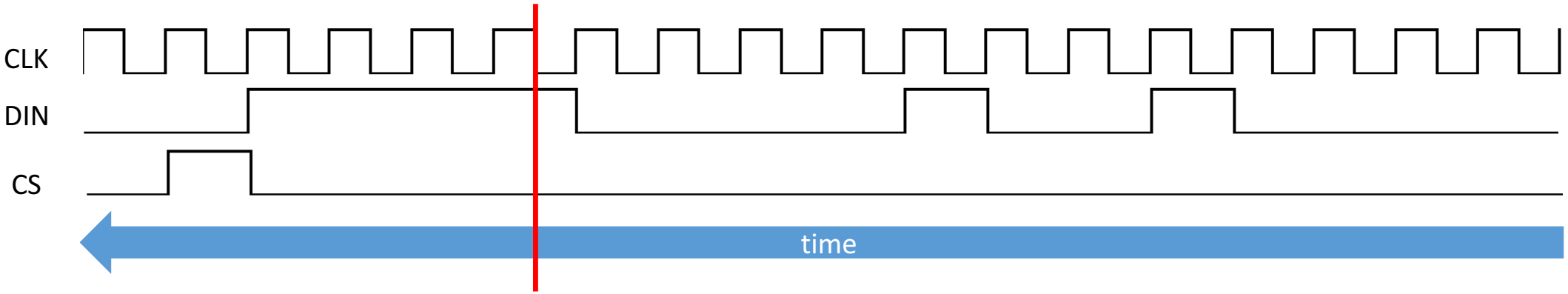
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	ADDRESS				DATA							
X	X	X	X	X	0	0	0	0	1	0	0	1	0	0	0



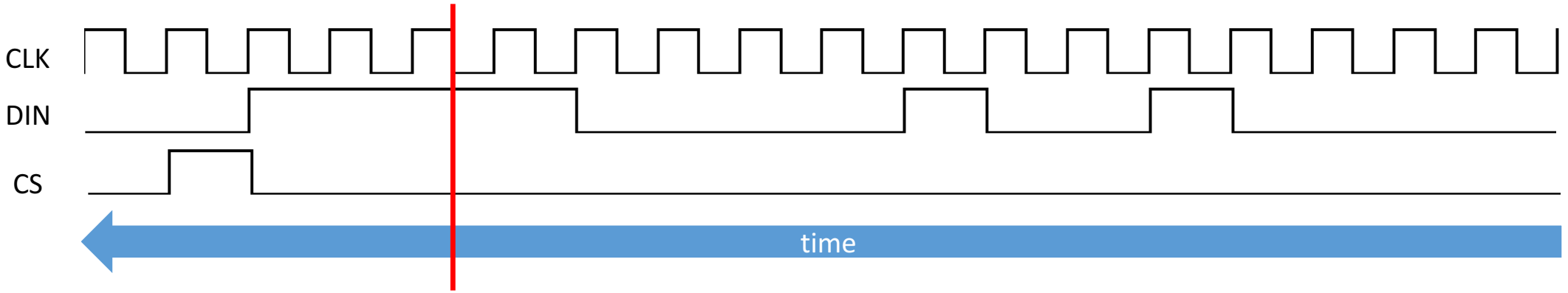
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	ADDRESS				DATA							
X	X	X	X	0	0	0	0	1	0	0	1	0	0	0	0



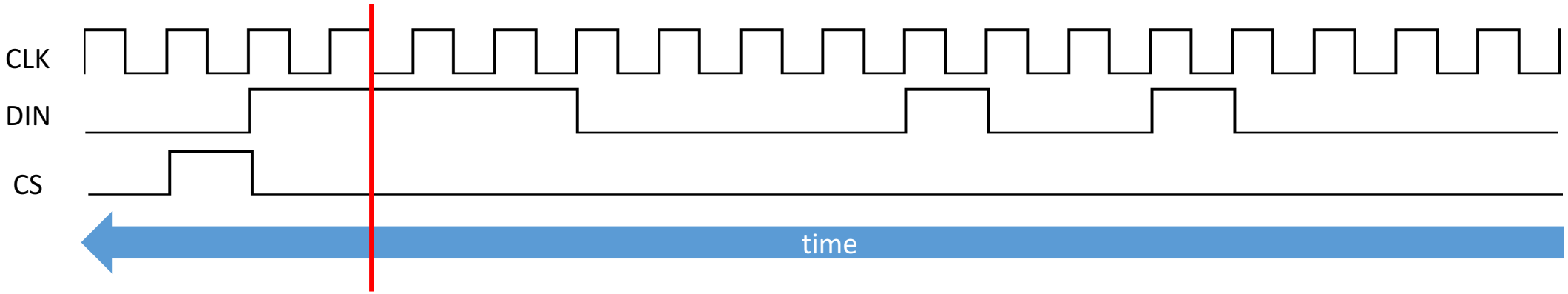
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	ADDRESS				DATA							
X	X	X	0	0	0	0	1	0	0	1	0	0	0	0	1



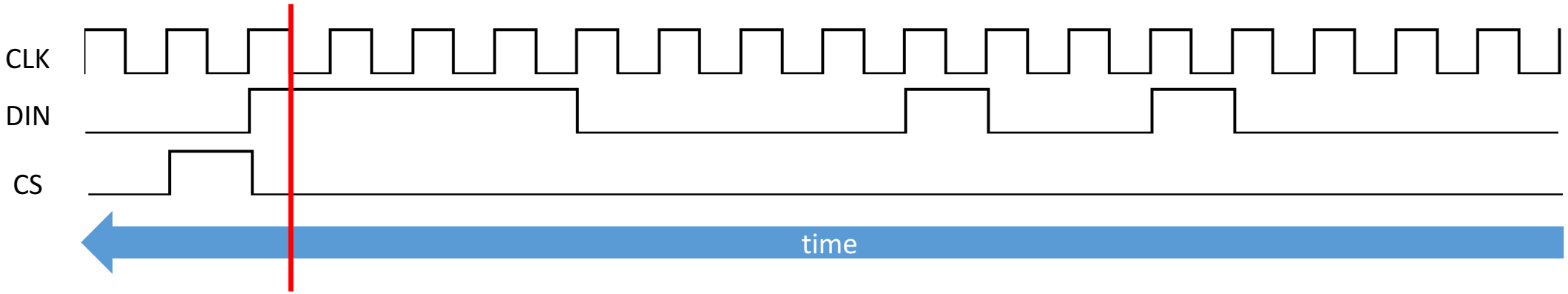
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	ADDRESS				DATA							
X	X	0	0	0	0	1	0	0	1	0	0	0	0	1	1



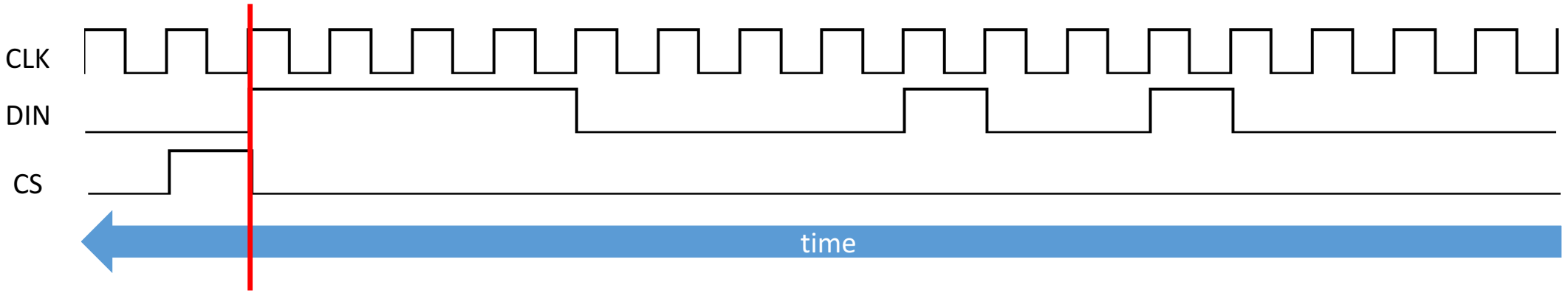
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	ADDRESS				DATA							
X	0	0	0	0	1	0	0	1	0	0	0	0	1	1	1



D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	ADDRESS				DATA							
0	0	0	0	1	0	0	1	0	0	0	0	1	1	1	1



D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X	ADDRESS				DATA							
0	0	0	0	1	0	0	1	0	0	0	0	1	1	1	1



Lab 5.1

- Lab5.1: Max7219與7-Seg LED練習—without code B decode mode
  - 將stm32的3.3V接到7-Seg LED板的VCC，GND接到GND，並選擇三個GPIO接腳分別接到DIN、CS和CLK。
  - 利用GPIO控制Max7219並在7-Seg LED上顯的第一位依序顯示0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, b, C, d, E, F (時間間隔1秒)，範例影片如下：

```
arr: .byte 0x7e, 0x30,
0x6d, …

main:
    BL  GPIO_init
    BL  max7219_init

    ldr r9, =arr
    ldr r2, =#0
.for_loop:
    mov r0, #1
    ldrb r1, [r9, r2]
    BL MAX7219Send
    BL Delay
    add r2, r2, #1
    cmp r2, #16
    bne .for_loop
    mov r2, #0
    b .for_loop
```

```
.equ DECODE_MODE,                    0x09

max7219_init:
       push {r0, r1, r2, lr}
       ldr r0, =#DECODE_MODE
       ldr r1, =#0x0
       BL MAX7219Send
       ldr r0, =#DISPLAY_TEST
       ldr r1, =#0x0
       BL MAX7219Send
       ldr r0, =#SCAN_LIMIT
       ldr r1, =0x0
       BL MAX7219Send
       ldr r0, =#INTENSITY
       ldr r1, =#0xA
       BL MAX7219Send
       ldr r0, =#SHUTDOWN
       ldr r1, =#0x1
       BL MAX7219Send
       pop {r0, r1, r2, pc}
```

```
.equ DATA,                                  0x20 //PA5
.equ LOAD,                                  0x40 //PA6
.equ CLOCK,                                 0x80 //PA7

MAX7219Send://input parameter: r0 is address , r1 is data
...
lsl r0, r0, #8
add r0, r0, r1
ldr r1, =#GPIOA_BASE
ldr r2, =#LOAD
ldr r3, =#DATA
ldr r4, =#CLOCK
ldr r5, =#GPIO_BSRR_OFFSET
ldr r6, =#GPIO_BRR_OFFSET
mov r7, #16//r7 = i
```

```
.max7219send_loop:
      mov r8, #1
      sub r9, r7, #1
      lsl r8, r8, r9 // r8 = mask
      str r4, [r1,r6]//HAL_GPIO_WritePin(GPIOA, CLOCK, 0);
      tst r0, r8
      beq .bit_not_set//bit not set
      str r3, [r1,r5]
      b .if_done
.bit_not_set:
      str r3, [r1,r6]
.if_done:
      str r4, [r1,r5]
      subs r7, r7, #1
      bgt .max7219send_loop
      str r2, [r1,r6]
      str r2, [r1,r5]
...
```

Lab 5.2

- Lab5.2: Max7219與7-Seg LED練習—use code B decode mode
  - 利用GPIO控制Max7219並在7-Seg LED上顯示自己的學號，例如學號為1234567則顯示下圖：
  - 完成以下程式碼，將放在student_id array 裡的學號顯示到7-seg LED上。

```
main:
    BL  GPIO_init
    BL  max7219_init


       ldr r9, =arr
       ldr r2, =#0
       ldr r3, =#8
       ldr r4, =#9
.for_loop:
       ldrb r1, [r9, r2]
       add r0, r2, #1
       sub r0, r4, r0
       BL MAX7219Send
       add r2, r2, #1
       cmp r2, #8
       bne .for_loop
loop:
       b loop
```

Lab 5.3

- Lab5.3 Max7219與7-SEG LED練習—顯示Fibonacci數
  - 請設計一組語程式偵測實驗板上的User button，當User button按N次時7-Seg LED上會顯示fib(N)的值。User button長按1秒則將數值歸零，範例影片如下：

Reference

- https://www.sparkfun.com/datasheets/Components/General/COM-09622-MAX7219-MAX7221.pdf