ARM
LPC2148 Architecture

Dr. T.G.Panse

Assistant Professor

Department of Electronics Engineering,

Yeshwantrao Chavan College of
Engineering,

Nagpur – 441 110

LPC2148

- Provided by NXP Semiconductor Ltd.
- Based on ARM v4T architecture version,
- Based on ARM7-TDMI processor
- Integrated Real-Time Emulation and Embedded Trace support.

  - Features of LPC2148 Family
  - LPC2148 – Internal architecture diagram
  - Pin diagram, Pin connect block,
  - Memory map
  - GPIO, PLL
  - Interrupt structure, Vectored Interrupt Controller (VIC)
  - Peripherals: ADC, DAC, Timer, PWM,
  - Serial Communication Interfaces:

LPC2148 Specifications

Processor	Architecture	: ARM v4T
	Processor	: ARM7-TDMI-S
	Instructions	: 32-bit ARM and 16-bit Thumb
	Debug support	: RT EmbeddedICE, Embedded Trace interface
Static RAM (On-chip)	Size	: 32 KB
	Additional	: 8 KB for USB DMA
Flash Program Mem. (On-chip)	Size	: 512 KB
	Programming	: ISP/IAP via on-chip boot-loader program.
GPIO	No. of pins	: up to 45 (fast GPIO lines, 5V tolerant)
	Features	: Configurable to fast GPIO
External Interrupts	Interrupts No.	: Four
	of pins	: Nine
	Sensitivity	: Rising/falling edge or low/high level sensitive
Timer/Counter	No. of Timers	: Two, 32-bit
	Operation	: Counter or timer operation
	Compare & Cap	: Four channels for each timer

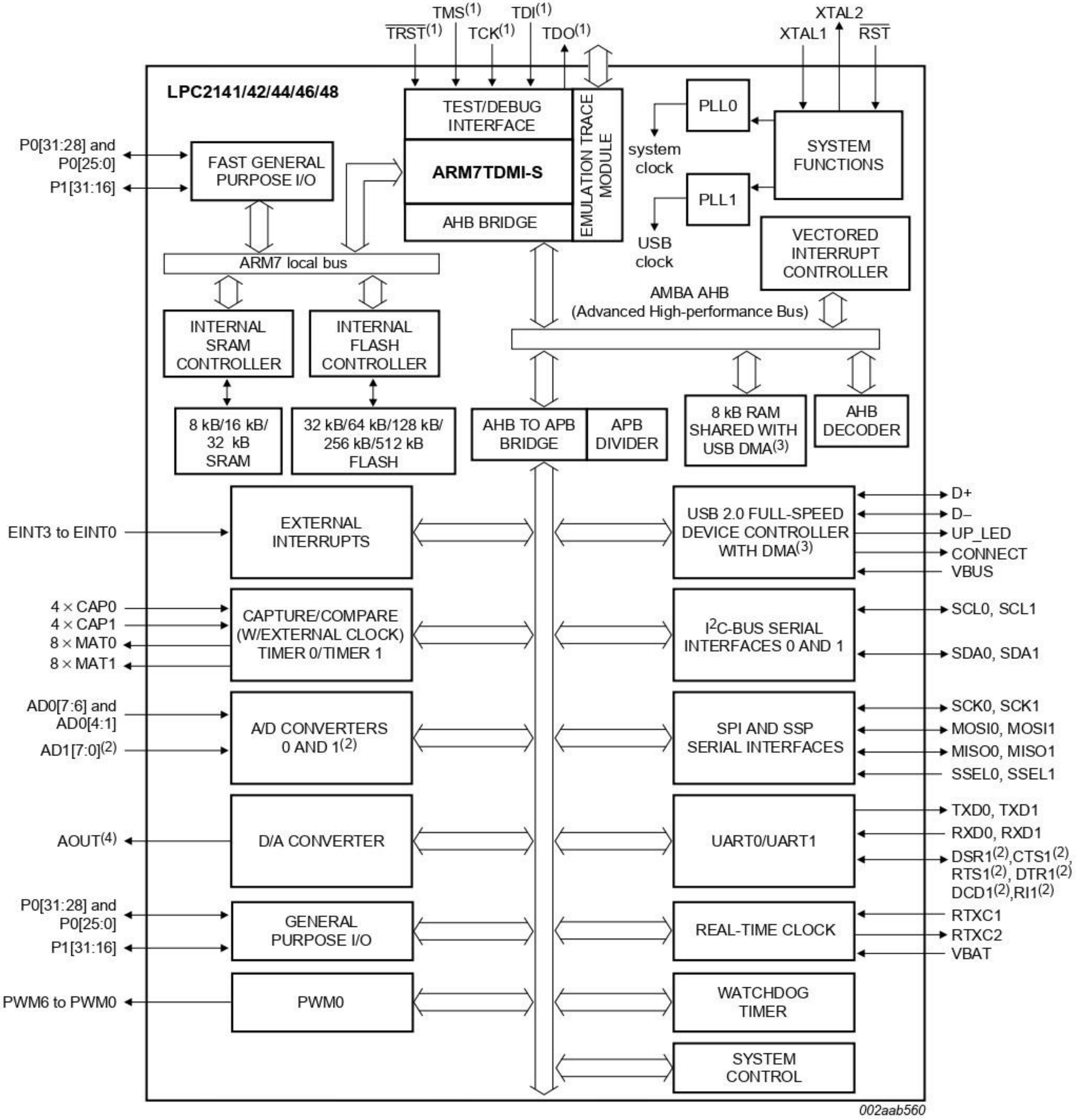
LPC2148 Specifications

PWM	No. of outputs If not enabled	: 6 single edge / 3 double edge controlled or mix : Used as a standard 32-bit timer/counter
Watchdog Timer	Timer	▪ 32-bit counter, divide by 4 fixed pre-scaler ▪ Internally resets the chip
RTC	Features	▪ Maintains calendar, clock, ▪ Provides Seconds, Minutes, Hours, day of week, day of Month, Month, day of Year, Year. ▪ Consumes very low power, ▪ Dedicated power pin, can use battery ▪ *Uses dedicated 32 kHz clock*
ADC	No. of ADC No. of channel Resolution Conversion time VREF	: Two, Successive approximation type : 14 (ADC0 – 6 channels, ADC1 – 8 channels) : 10-bit : 2.44 ms per channel : >2.5V, <3.3 V
DAC	No. of DAC Resolutio n	: One : 10-bit : 3.3 V

LPC2148 Specifications

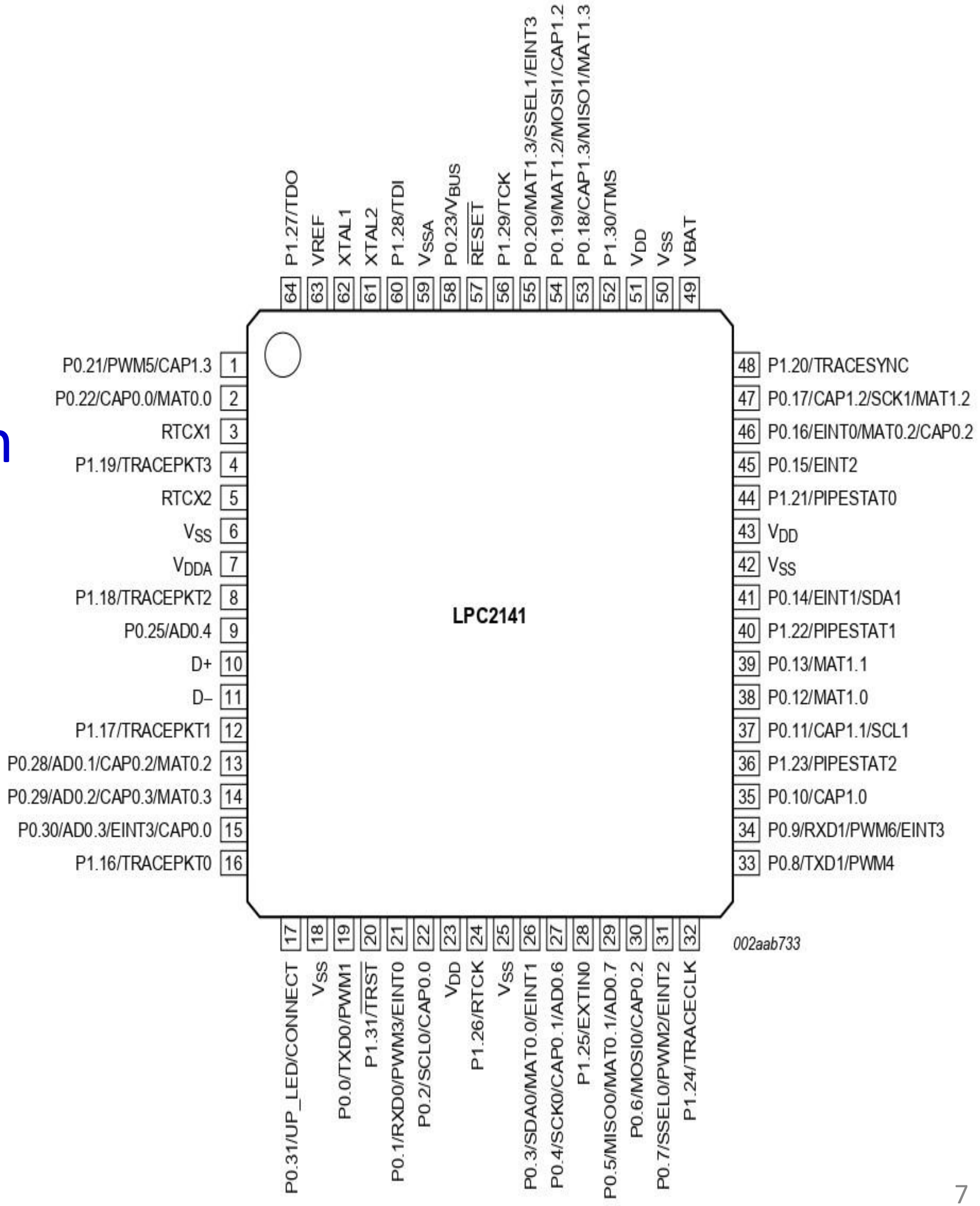
UART	No. of channels	: Two (16C550 compliant), UART0 and UART1
	Modem Interface	: UART1 with full modem interface
SPI	No. of channels	: One : Full Duplex, Multiple master and slaves support
SSP	Compatibility	: Motorola SPI, 4-wire TI SSI, and National Semiconductor Microwire
I2C	No. of buses	: Two
USB	USB	: One
	ports	: USB 2.0
	Complian	: Full-speed (12 Mbps)
	t	: USB Device
	Data transfer	: 2 KB endpoint RAM
	rate Controller	
	Special feature	
Package		: LQFP64 (Low profile Quad Flat Package)
Features		▪ 128-bit memory interface 60 MHz operation

LPC2148 Architectu



LPC2148 Pins and Si

- LQFP64 package,
- 64 (physical) pins,
- Multiple functions assigned
- By default I/O function
- Power supply, ground, osc pins not multiplexed



LPC2148 I/O pins

- 64 pins are attached to two 32-bit I/O ports, Port-0 & Port-1

- Port-0 pins are designated as P0.0 – P0.31
- Port-1 pins P1.0 - P1.31

- Pins P0.24, P0.26, P0.27, P1.0-P1.15 are unavailable.

- Pin functions are multiplexed, up to 4 functions assigned to each pin.
  - Port-0 pins multiplex peripheral pin, & comm. interface pin functions
  - Port-1 pins multiplex JTAG interface, Trace function

Advantages: keeps size small, adds more functionalities to devices

Disadvantages: if functions not carefully selected, some can’t be

availed

- Pin function select Registers: PINSEL0, PINSEL1, PINSEL2

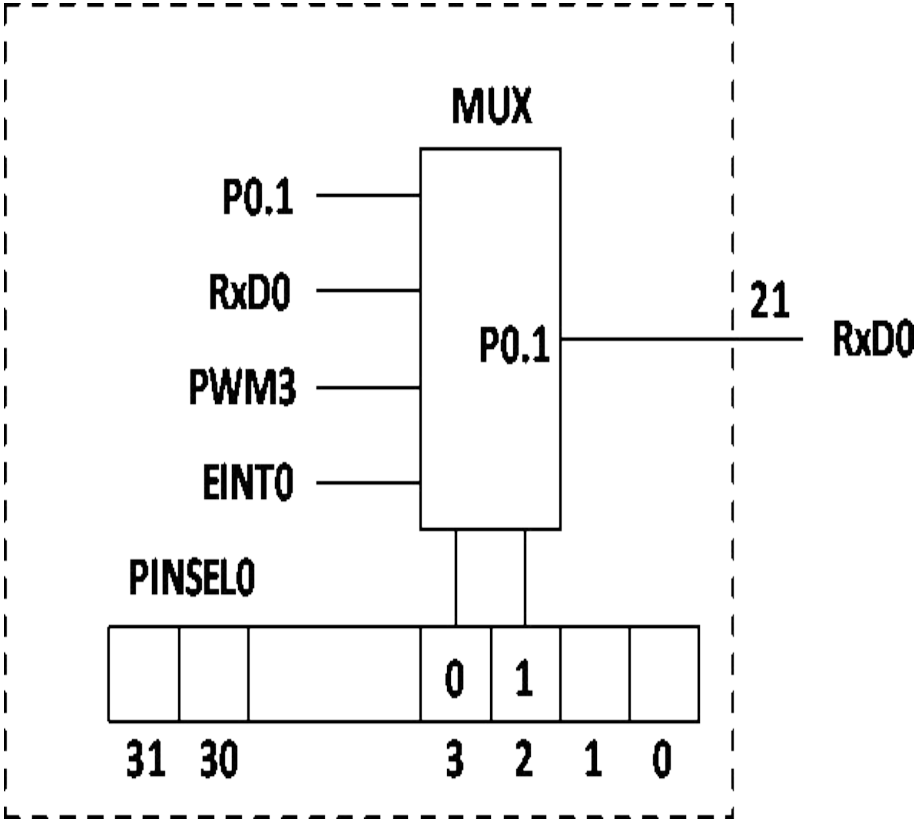
PINSEL0 selects functions of pins P0.0 to P0.15,

PINSEL1 selects functions of pins P0.16 to P0.31

- PINSEL2 selects functions of pins P1.16 to P1.31

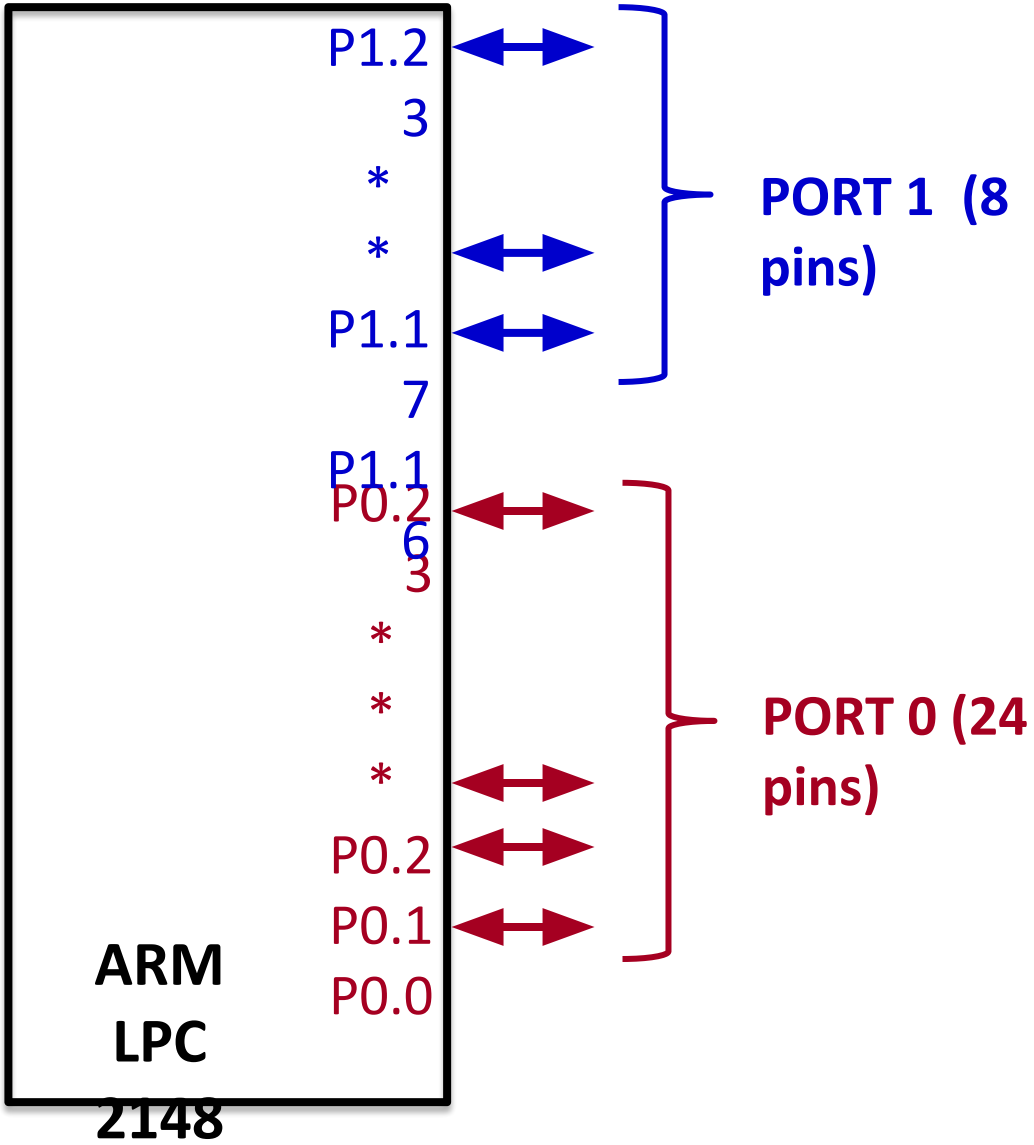
LPC2148 I/O pins

- Pin functions are multiplexed, up to 4 functions assigned to each pin.
- Port-0 pins multiplex peripheral pin, & comm. interface pin functions



LPC2148 GPIO pins

Port 0 (P0.23---P0.0) and Port1 (P1.23-----P1.16)



P1.23

*

*

P1.17

P1.16

P0.23

*

*

*

P0.2

P0.1

P0.0

PORT 1 (8 pins)

PORT 0 (24 pins)

ARM LPC 2148

LPC2148 GPIO registers

There are 4 Slow GPIO registers :

1 **IOxDIR (GPIO Port Direction control register) :** This is a 32-bit wide register. This register individually controls the direction of each port pin. Setting a bit to **‘1’** configures the corresponding pin as an **output pin**. Setting a bit to **‘0’** configures the corresponding pin as an **input pin.**

2. **IOxPIN (GPIO Port Pin value register):** This is a 32-bit wide register. This register is used to read/write the value on Port (PORT0/PORT1). But care should be taken while writing. Masking should be used to ensure write to the desired pin.

3. **IOxSET (GPIO Port Output Set register) :** This is a 32-bit wide register. This register is used to make pins of Port (PORT0/PORT1) HIGH. **Writing one to specific bit makes that pin HIGH. Writing zero has no effect.**

4. **IOxCLR (GPIO Port Output Clear register) :** This is a 32-bit wide register. This register is used to make pins of Port LOW. **Writing one to specific bit makes that pin LOW. Writing zeroes has no effect.**

LPC2148 GPIO Registers

There are 4 Slow GPIO registers :

1 **IOxDIR (GPIO Port Direction control register) :** This is a 32-bit wide register. This register individually controls the direction of each port pin. Setting a bit to **‘1’** configures the corresponding pin as an **output pin**. Setting a bit to **‘0’** configures the corresponding pin as an **input pin.**

IO**0**DIR is used for **PORT0** pins to configure as Input or Output pins

IO**1**DIR is used for **PORT1** pins to configure as Input or Output pins

Example

a) Configure PORT 0 pins P0.7 to P0.4 as Output and other pins as Input

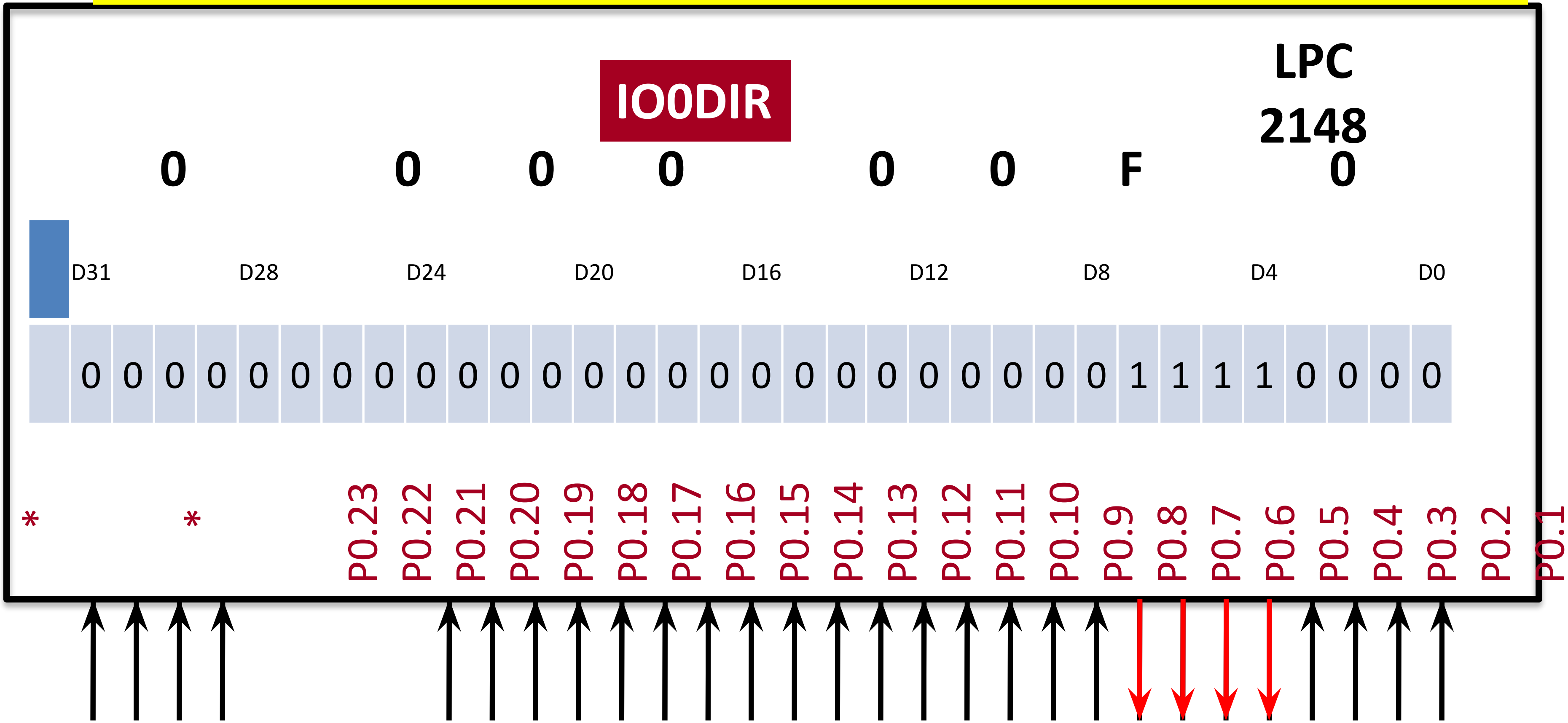
IO0DIR =0X000000F0; //P0.7 to P0.4 are now acting as a OUTPUT pins

a) Configure PORT 1 pins P1.23 to P1.16 as input and other pins as output

IO1DIR =0XFF00FFFF; //P1.23 to P1.16 are now acting as a

LPC2148 IOxDIR (GPIO Port Direction control register)

IO0DIR =0X000000F0; //P0.7 to P0.4 are now acting as a OUTPUT pins



IO0DIR

LPC 2148

0 0 0 0 0 0 F 0

D31 D28 D24 D20 D16 D12 D8 D4 D0

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

LPC2148   GPIO registers

2. **IOxPIN (GPIO Port Pin value register):** This is a 32-bit wide register. This register is used to read/write the value on Port (PORT0/PORT1). But care should be taken while writing. Masking should be used to ensure write to the desired pin.

Examples :

a) Writing F to P0.7-P0.4

```
IO0PIN = IO0PIN | (0x000000F0)
```

b) Writing 1 to P0.4 using IO0PIN

```
IO0PIN = IO0PIN | (1<<4)
```

c) Writing 0 to P0.4 using IO0PIN

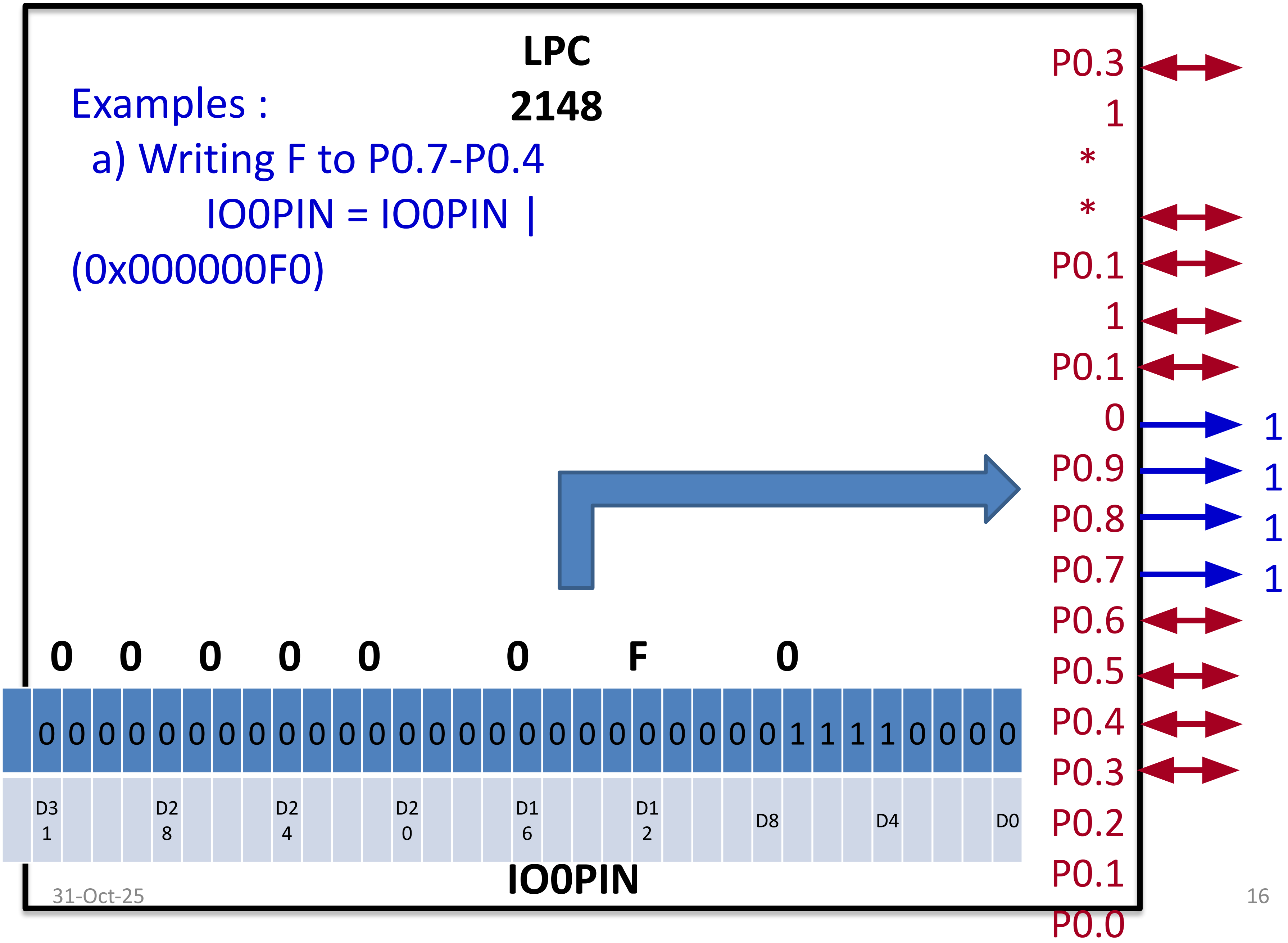
```
IO0PIN = IO0PIN & (~(1<<4) )
```

LPC2148 IOxPIN (GPIO Port Pin value register):

Examples :

a) Writing F to P0.7-P0.4

IO0PIN = IO0PIN |
(0x000000F0)



0 0 0 0 0 0 F 0

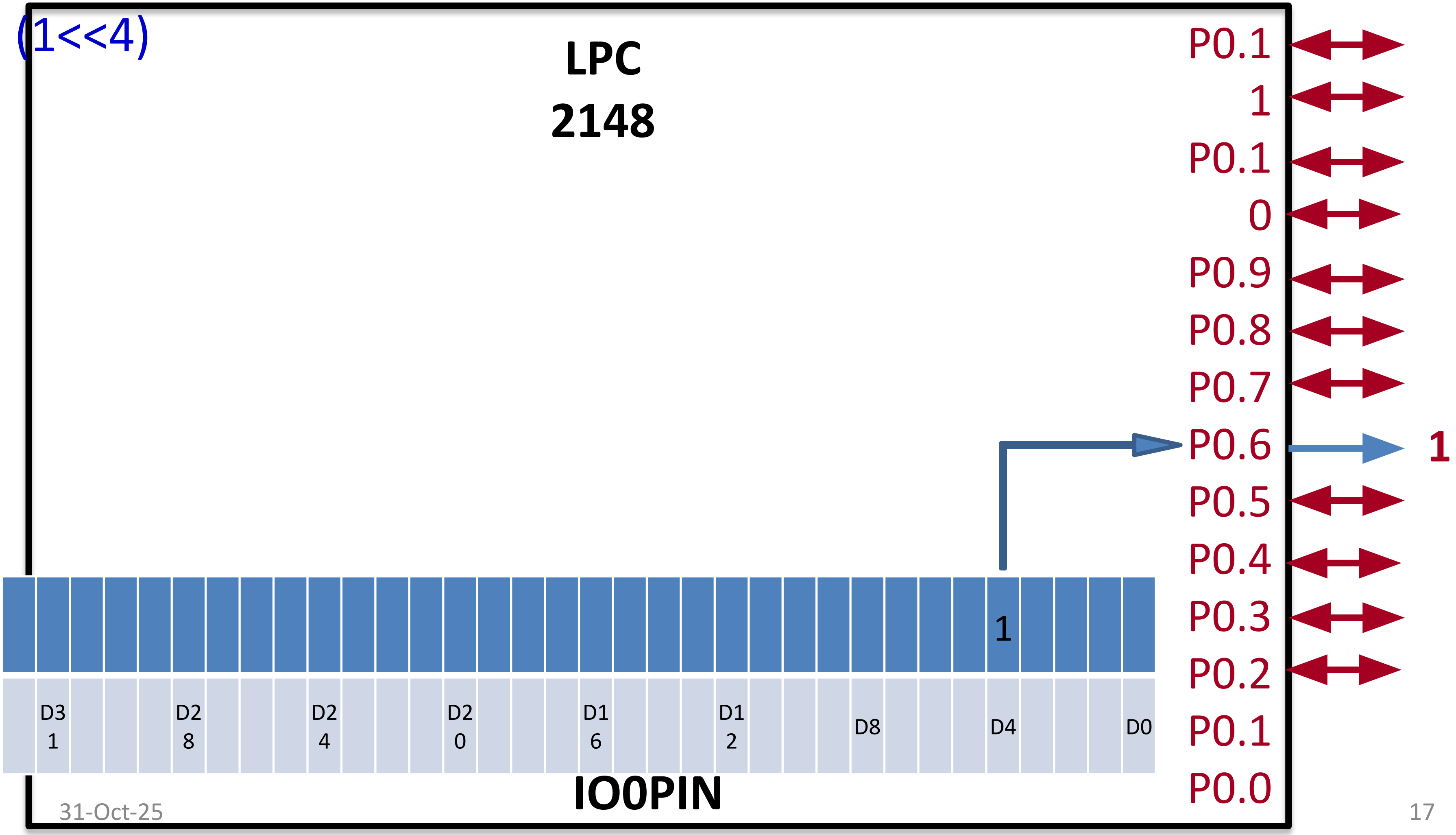
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
	D31				D28				D24				D20				D16				D12				D8				D4				D0

IO0PIN

LPC2148 IOxPIN (GPIO Port Pin value register):

c) Writing 1 to P0.4 using IO0PIN

IO0PIN = IO0PIN | (1<<4)



LPC 2148

P0.11

P0.10

P0.9

P0.8

P0.7

P0.6

1

P0.5

P0.4

P0.3

P0.2

P0.1

P0.0

1

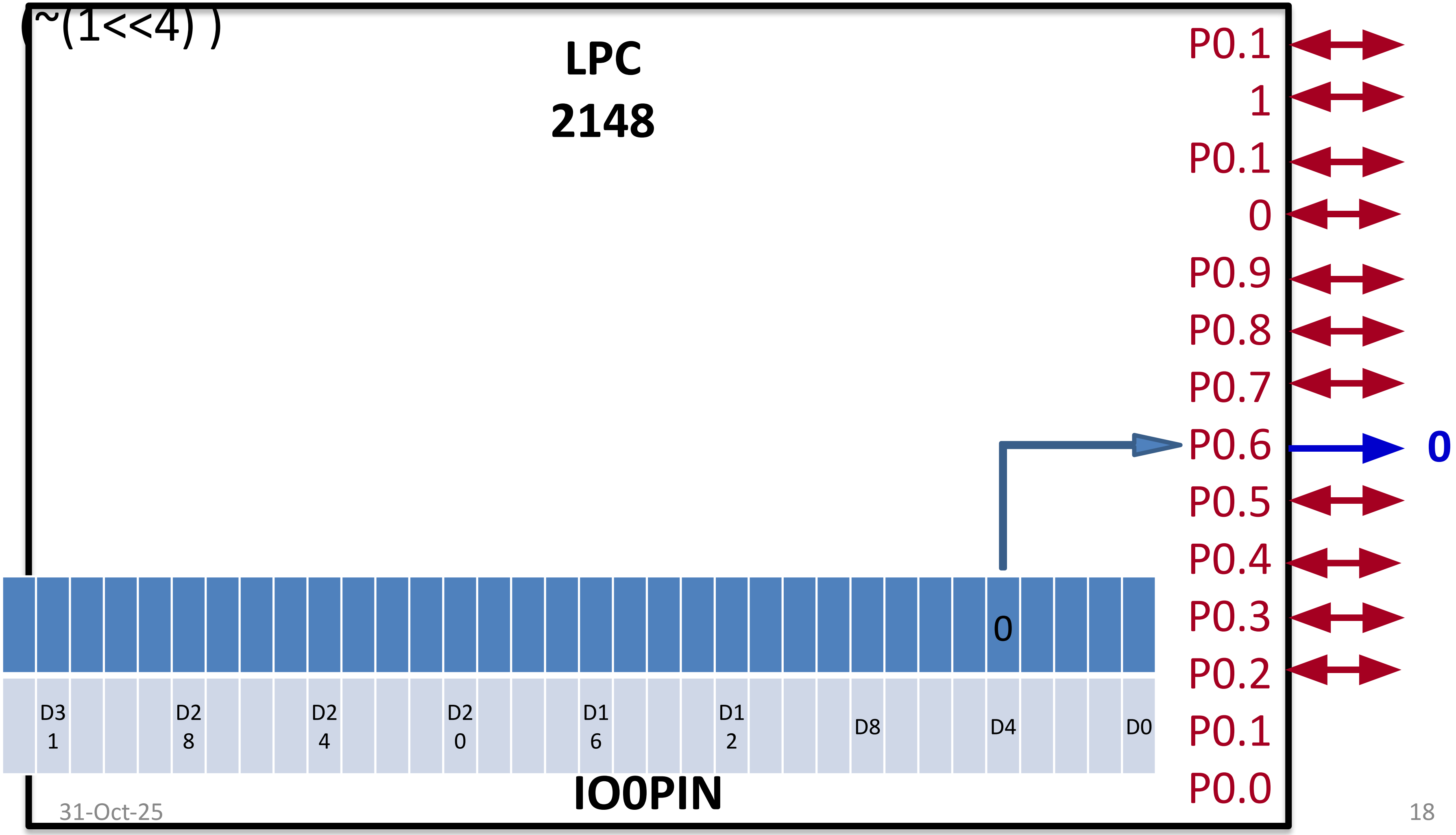
D31 D28 D24 D20 D16 D12 D8 D4 D0

IO0PIN

LPC2148 IOxPIN (GPIO Port Pin value register):

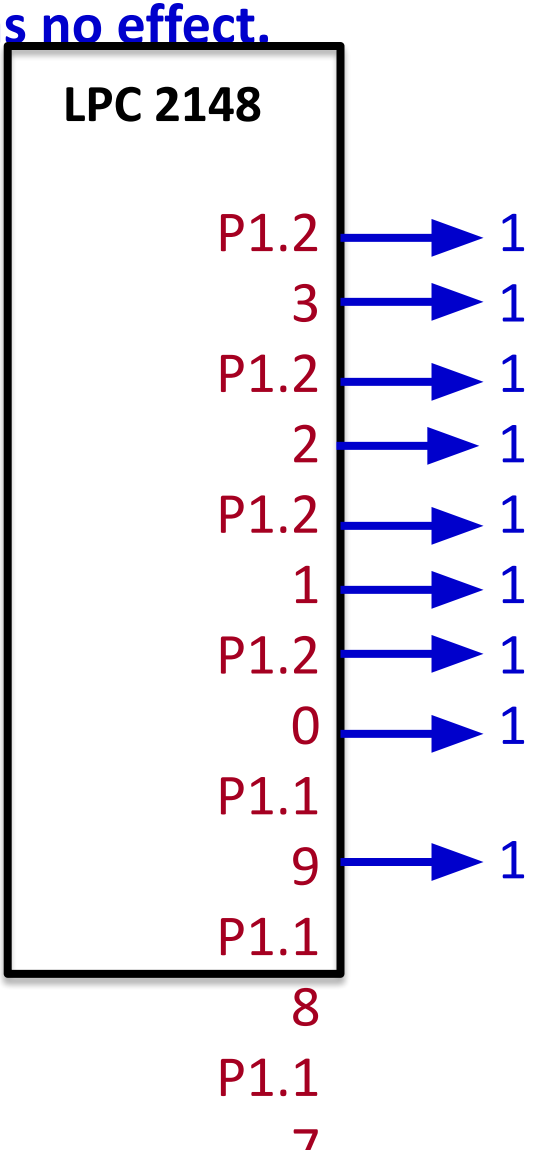
b) Writing 0 to P0.4 using IO0PIN

```
IO0PIN = IO0PIN & (~(1<<4) )
```



LPC2148 IOx SET GPIO registers

3. **IOxSET (GPIO Port Output Set register) :** This is a 32-bit wide register. This register is used to make pins of Port (PORT0/PORT1) HIGH. **Writing one to specific bit makes that pin HIGH. Writing zero has no effect.**



Examples :

a) Set pin P1.16 to P1.23 HIGH

```
IO1SET = 0x00FF0000;
```

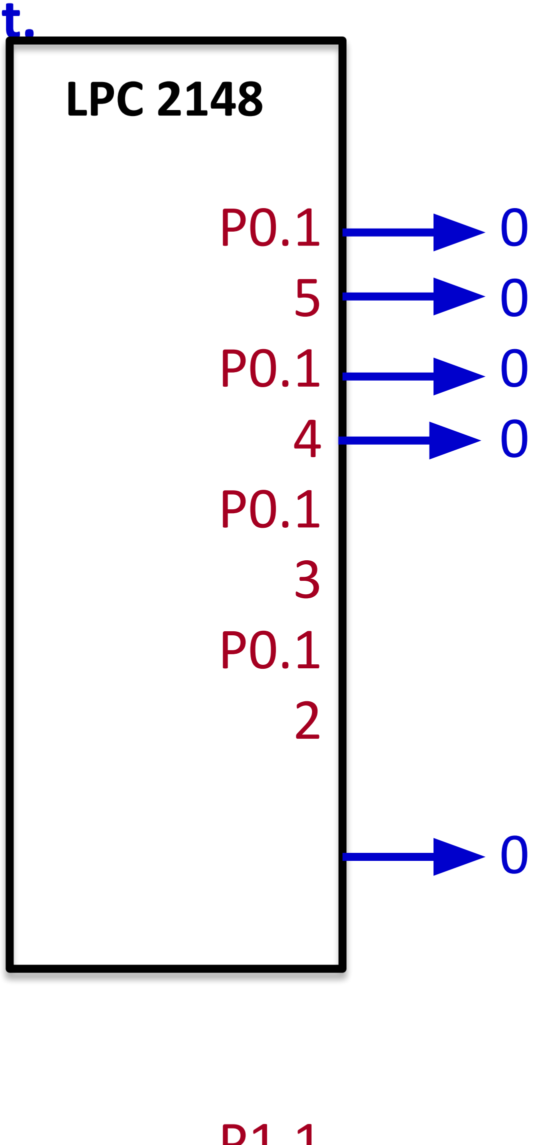
Examples :

b) Set pin P0.4 HIGH.

```
IO0SET = (1<<4);
```

LPC2148 IOx SET GPIO registers

4. **IOxCLR (GPIO Port Output Clear register) :** This is a 32-bit wide register. This register is used to make pins of Port LOW. **Writing one to specific bit makes that pin LOW. Writing zeroes has no effect.**



Examples :

a) Set pin P0.8 to P0.15 LOW

IO0CLR = 0000F000;

Examples :

b) Set pin P1.16 LOW.

IO1CLR = (1<<16);

LPC2148 I/O pin – Examples using Keil

```
#include <lpc214x.h>
```

; This header file includes all files for LPC214x series of microcontrollers

```
IO0DIR = (1<<7);
```

; It uses the IODIR register and make the pin 7 of Port 0 as output.

```
IO0SET = (1<<7) ;
```

; This register sets the P0.7 to HIGH (Logic 1)

```
IO0CLR = (1<<7);
```

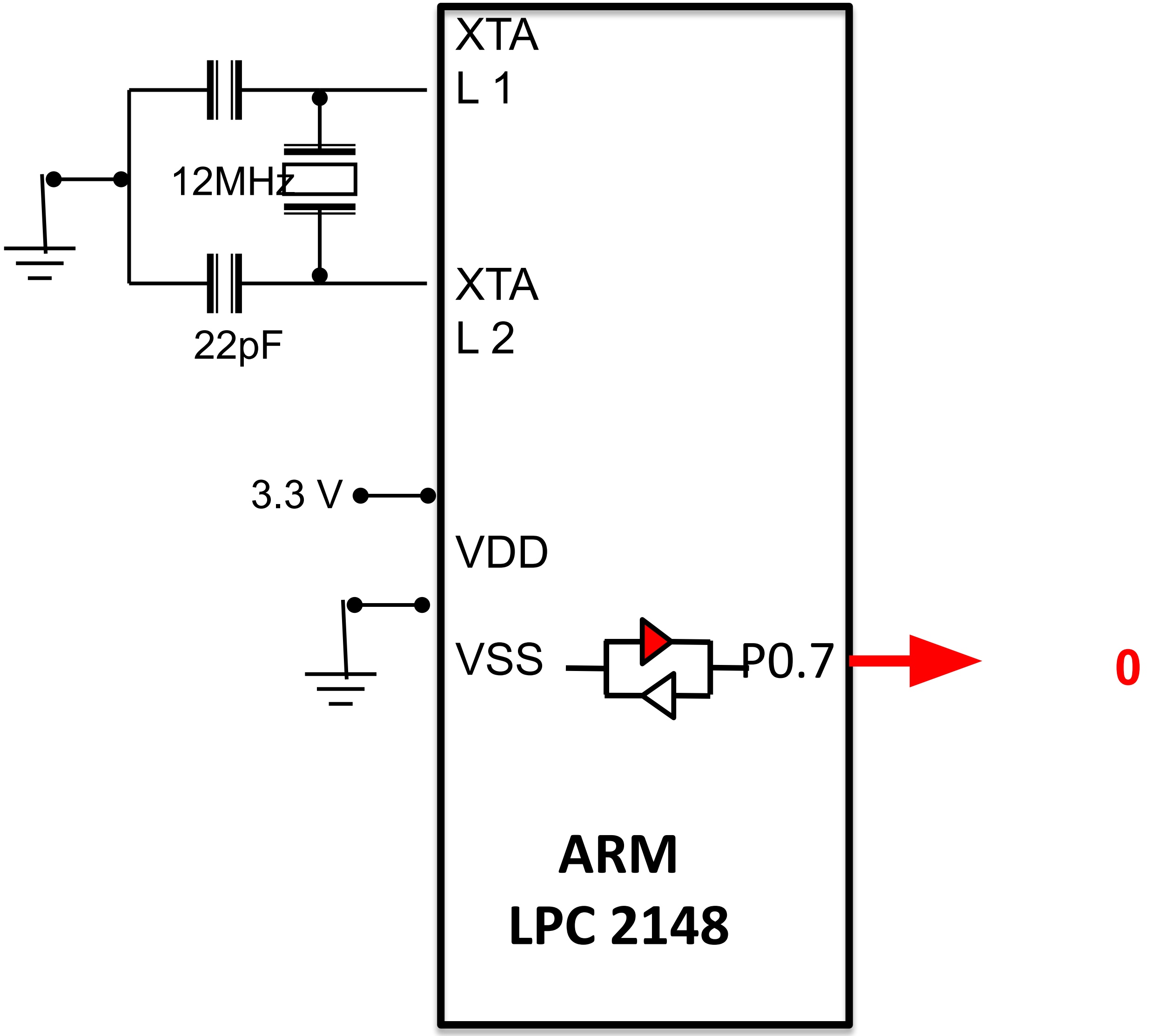
This register clears the P0.7 to LOW (Logic 0)

LPC2148 I/O pin – Examples using Keil

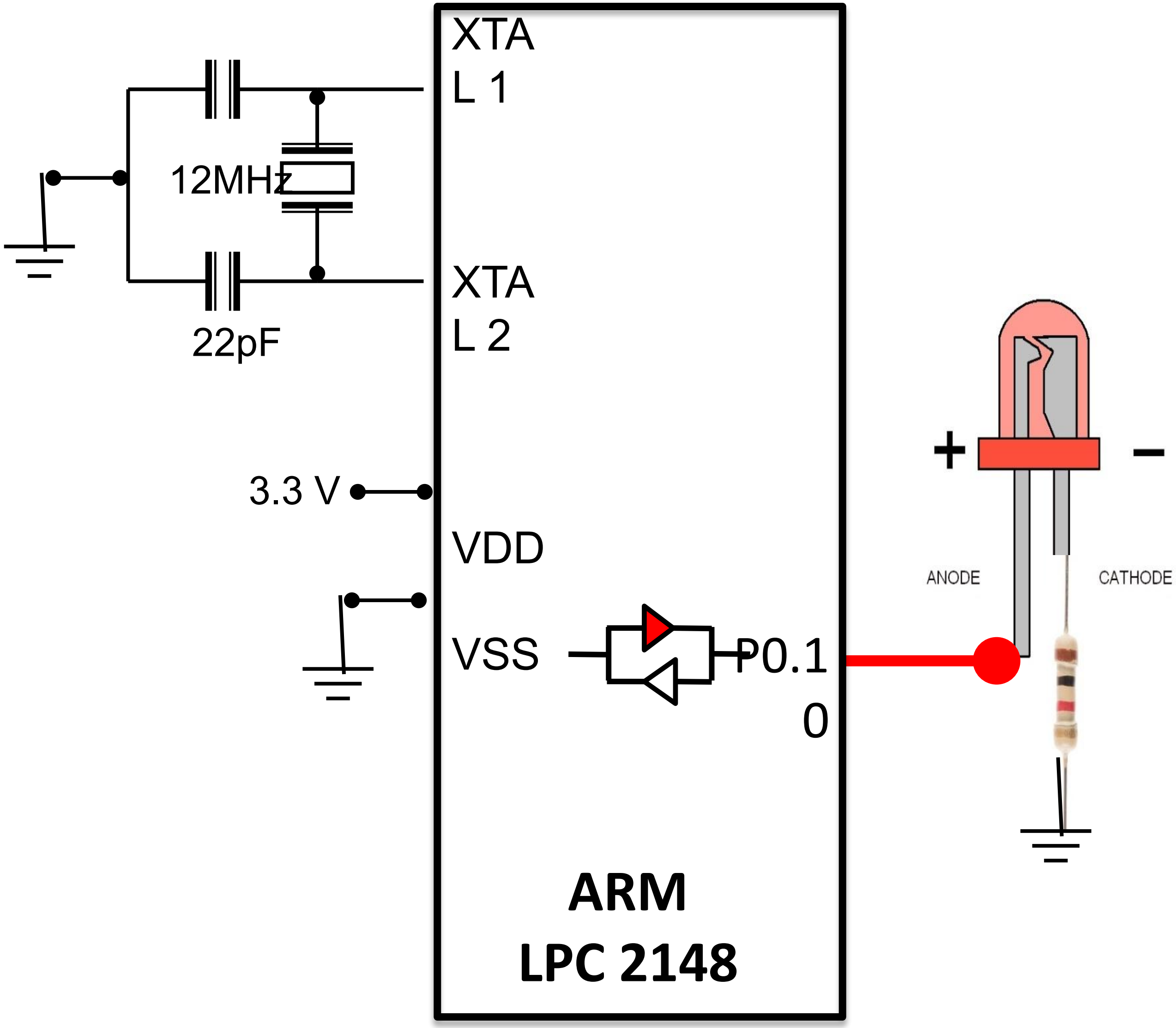
```
#include <lpc214x.h>  // This header file includes all files for
LPC214x main()
{
IO0DIR = (1<<7);                // make the pin 7 of Port 0 as
output.
IO0SET = (1<<7) ;          // This register sets the P0.7 to HIGH
(Logic 1)
IO0CLR = (1<<7);          //This register clears the P0.7 to LOW
(Logic 0)
}
```

LPC2148 I/O pin – Examples using Keil

Q. Write program to send logic 1 and 0 on P0.7



Q. Write program to blink LED connected to P0.10



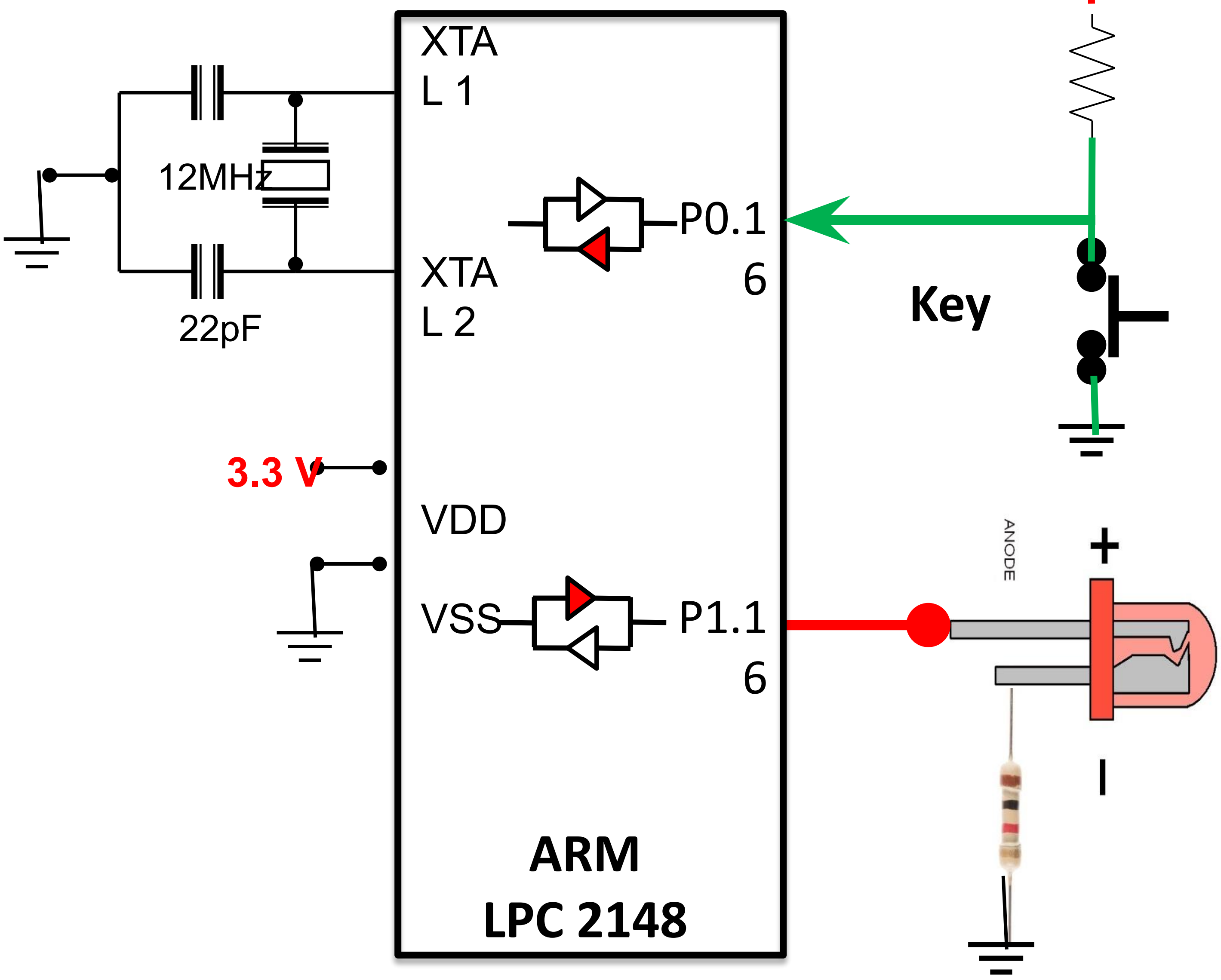
Q. Write program to blink LED connected to P0.10

```
#include <lpc214x.h>      //include header files for LPC-214x
series
void delay_ms(unsigned int count)
{
  unsigned int j=0,i=0;
  for(j=0;j<count;j++)     //For loop to create delay
  {
    for(i=0;i<3000;i++);
  }
}
```

Q. Write program to blink LED connected to P0.10

```
main ()
{
   IO0DIR = (1<<10);       //Configure the pin P0.10 as
OUTPUT;
  while(1)                  // While loop to execute program
continueously
   {
     IO0SET = (1<<10) ;    // Make the pin P0.10 HIGH
(LED ON)
       delay_ms(1000);
     IO0CLR = (1<<10);    // Make the pin P0.10 LOW
(LED OFF)
       delay_ms(1000);
   }
}
```

Q. Write program to turn ON LED if Switch is pressed, LED is Connected to P1.16 and Switch is connected to P0.16

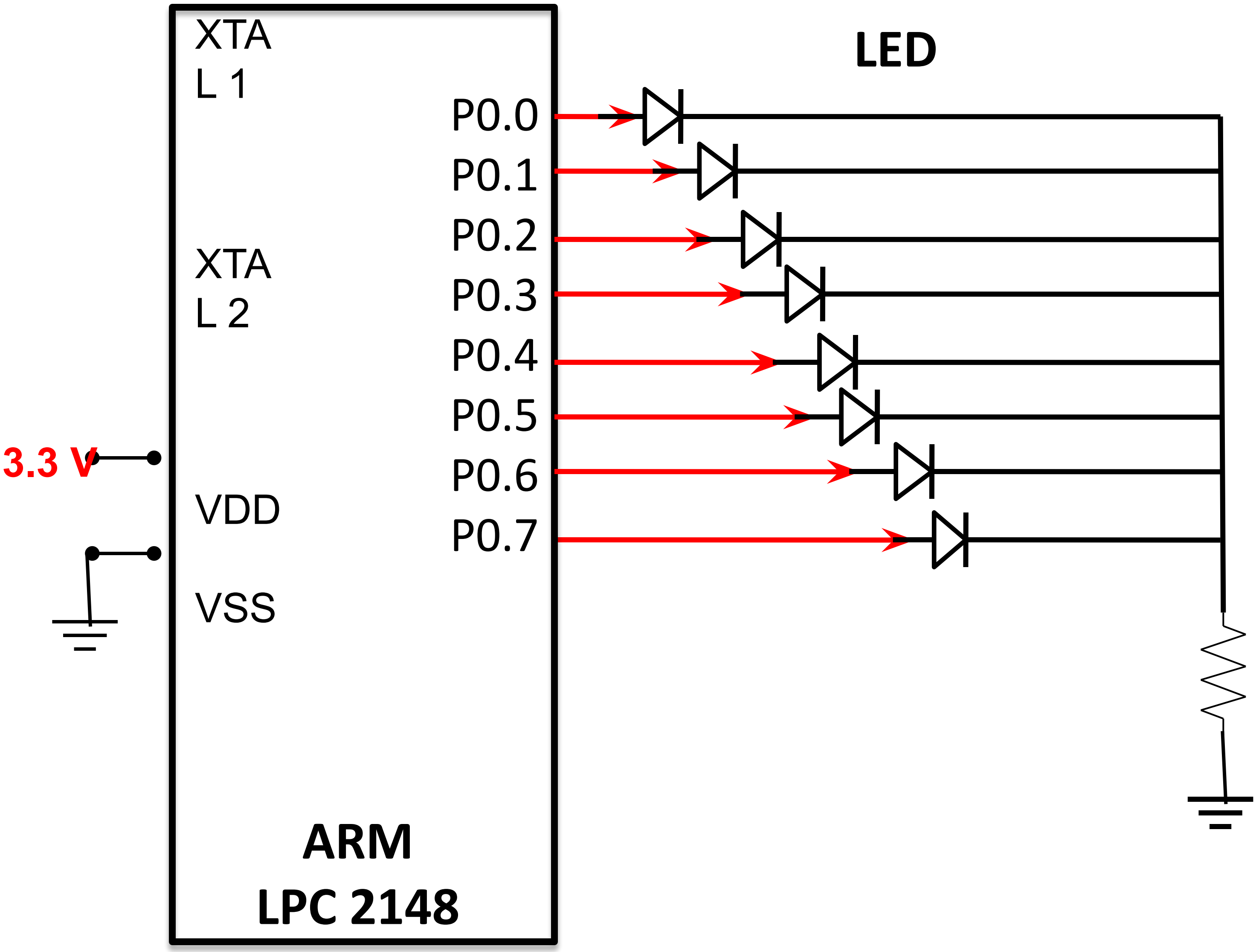


Q. Write program to turn ON LED if Switch is pressed, LED is connected to P1.16 and

Switch is connected to P0.16

```
#include<lpc214x.h>
void main()
{
    IO1DIR =0X00010000;                    //Port 1.16 is now acting as a
output pin
    IO0DIR = 0xFFFEFFFF;                        //Port 0.16 is now acting as a
input pin
    while(1)
    {
        if((IO0PIN & (1<<16)) ==0)     //Checking 16th pin of Port 0
(P0.16=0)
            IO1SET =0X00010000;   //Port 1.16 high now (LED is glowing)
        else
            IO1CLR =0X00010000;   //Port 1.16  low now (LED is OFF)
    }
}
```

Q. Interface 8 LED to P0.0 to P0.7 and write program to blink



Q. Interface 8 LED to P0.0 to P0.7 and write program to blink

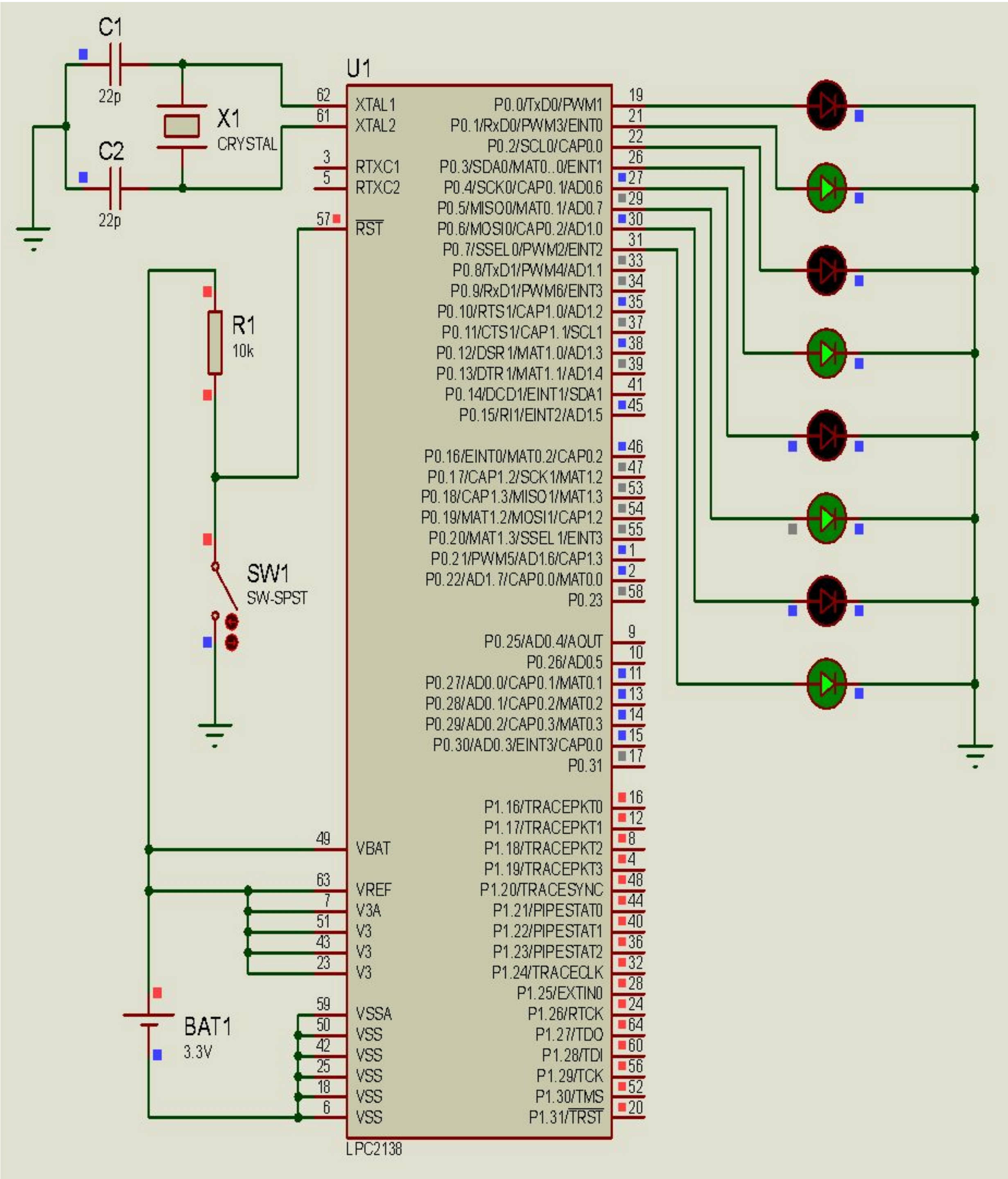
```
#include<lpc214x.h>
void delay();
void main()
{
IO0DIR | =0X000000FF;      //Port 0 is now acting as a output pin
while(1)
     {
IO0SET0 |=0X000000FF;      //Port 0's all pins are high now (LED is glowing)
delay();
IO0CLR0 |=0X000000FF;      //Port 0's all pins are low now (LED is OFF)
delay();
     }
}
void delay()
{
unsigned int i;
for(i=0;i<90000;i++);
}
```

Q. Interface 8 LED to P0.0 to P0.7 and write program to blink alternate LED

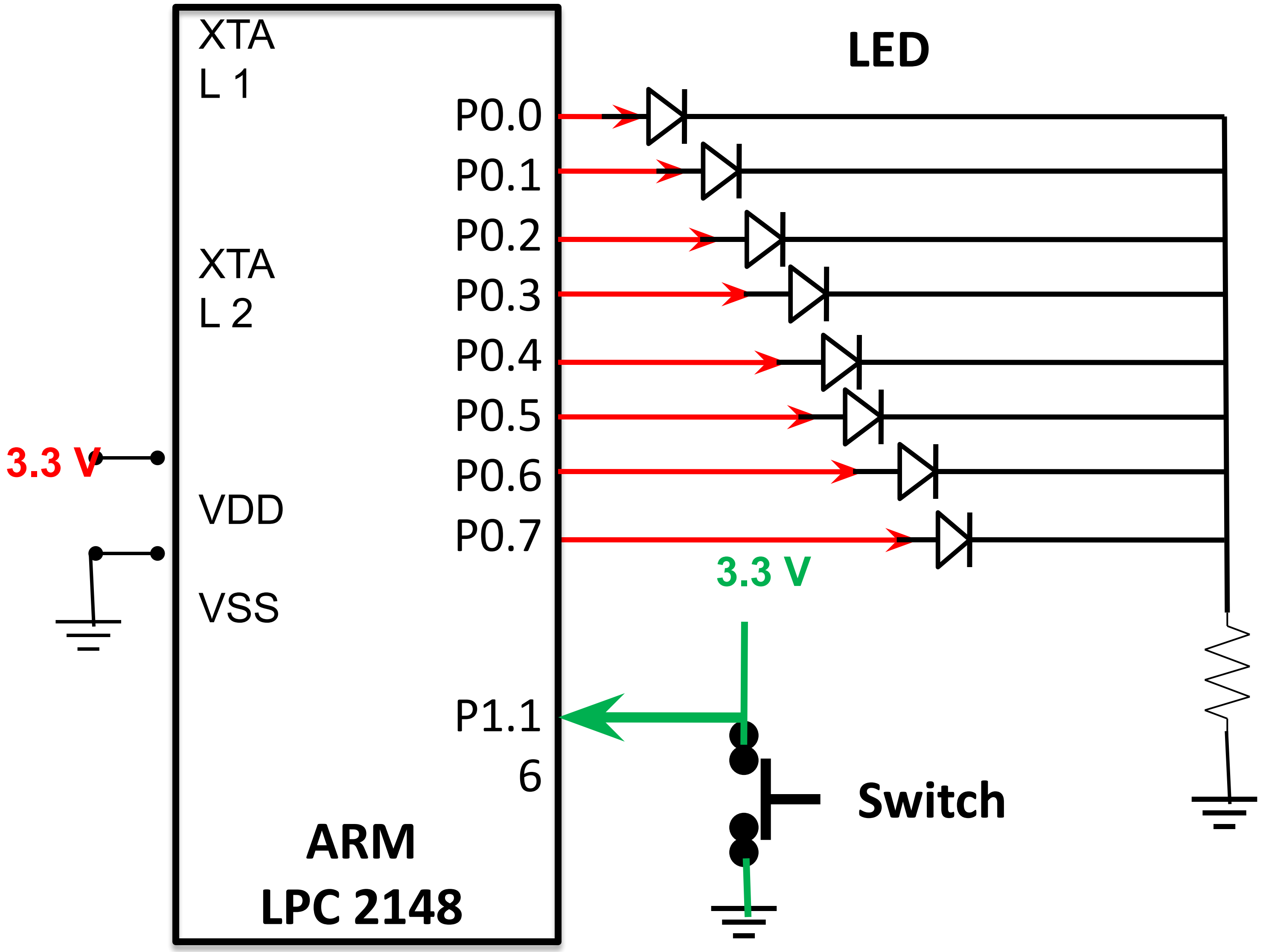
```
#include<lpc214x.h>
void delay();
void main()
{
IO0DIR |=0X000000FF;      //Port 0 is now acting as a output pin
while(1)
        {
IO0SET0 |=0X000000AA;   //Port 0's alternate pins are high now (LED is
glowing)
delay();
IO0CLR0 |=0X000000FF;    //Port 0's all pins are low now (LED is OFF)
IO0SET0 |=0X00000055;    //Port 0's alternate pins are high now (LED is
glowing)
delay();
IOCLR0 |=0X000000FF;      //Port 0's all pins are low now (LED is OFF)
        }
}
void delay()
{
unsigned int i;
for(i=0;i<90000;i++);
```

Q. Interface 8 LED to P0.0 to P0.7 and write program to blink alternate LED

Simulation using Proteus SW



Q. Interface 8 LED to P0.0 to P0.7 and Switch to P1.16

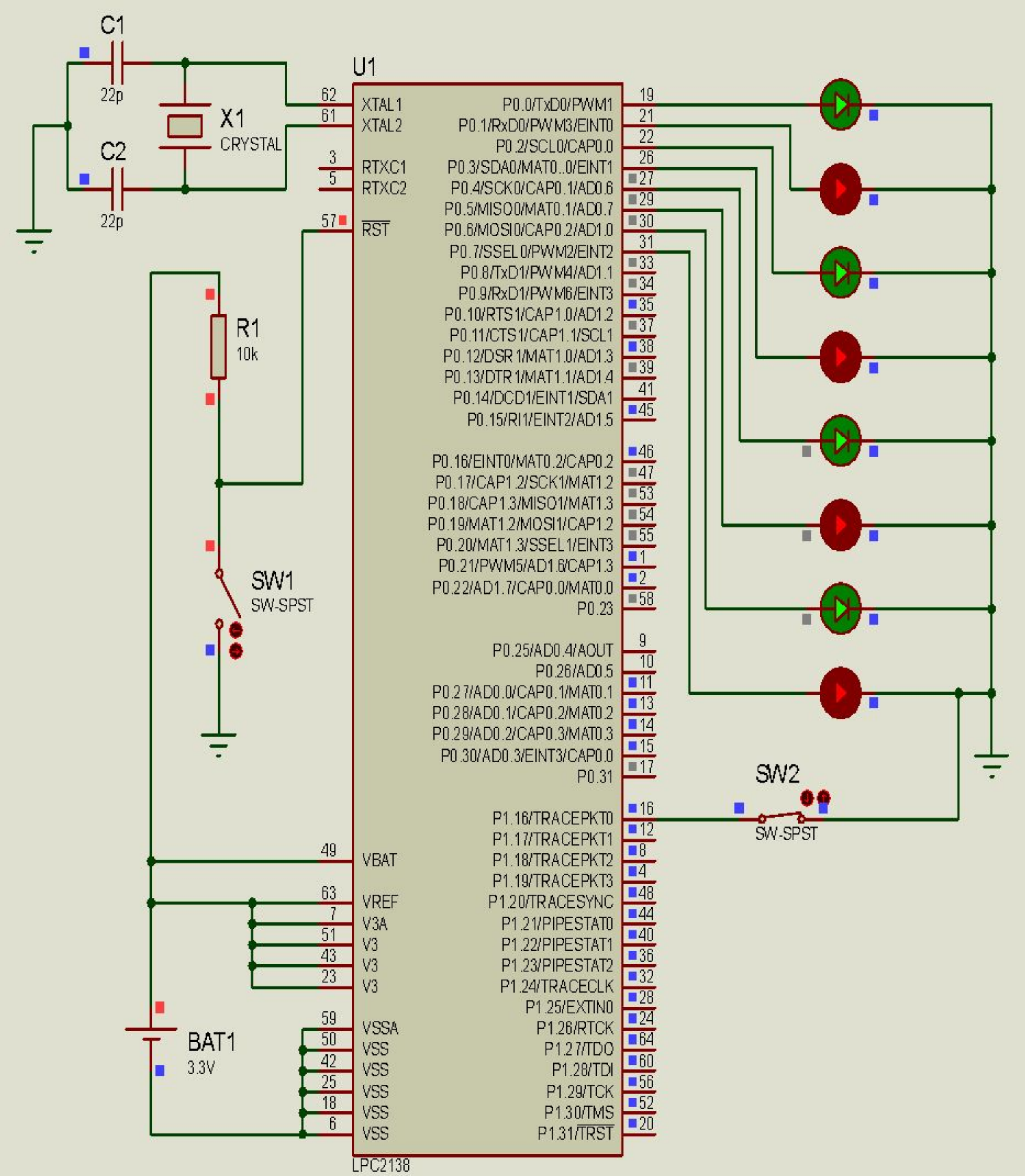


Q. Write program to turn ON LED’s if Switch is pressed, 8 LED are connected to P0.0 to P0.7 and Switch is connected to P1.16

```
#include<lpc214x.h>
void main()
{
   IO1DIR =0XFFFEFFFF;              //Port 1.16 is now acting as a input
pin
   IO0DIR = 0x000000FF;             //P0.0 to P0.7 are now acting as an
output pins
   while(1)
   {
      if((IO1PIN & (1<<16)) ==0)          //Checking 16th pin of Port 1
(P1.16=0)
         IO0SET =0X000000FF;        // P0.0 to P0.7 high now (LED is
glowing)
      else
         IO0CLR = 0X000000FF;       // P0.0 to P0.7 low now (LED is OFF)
   }
```

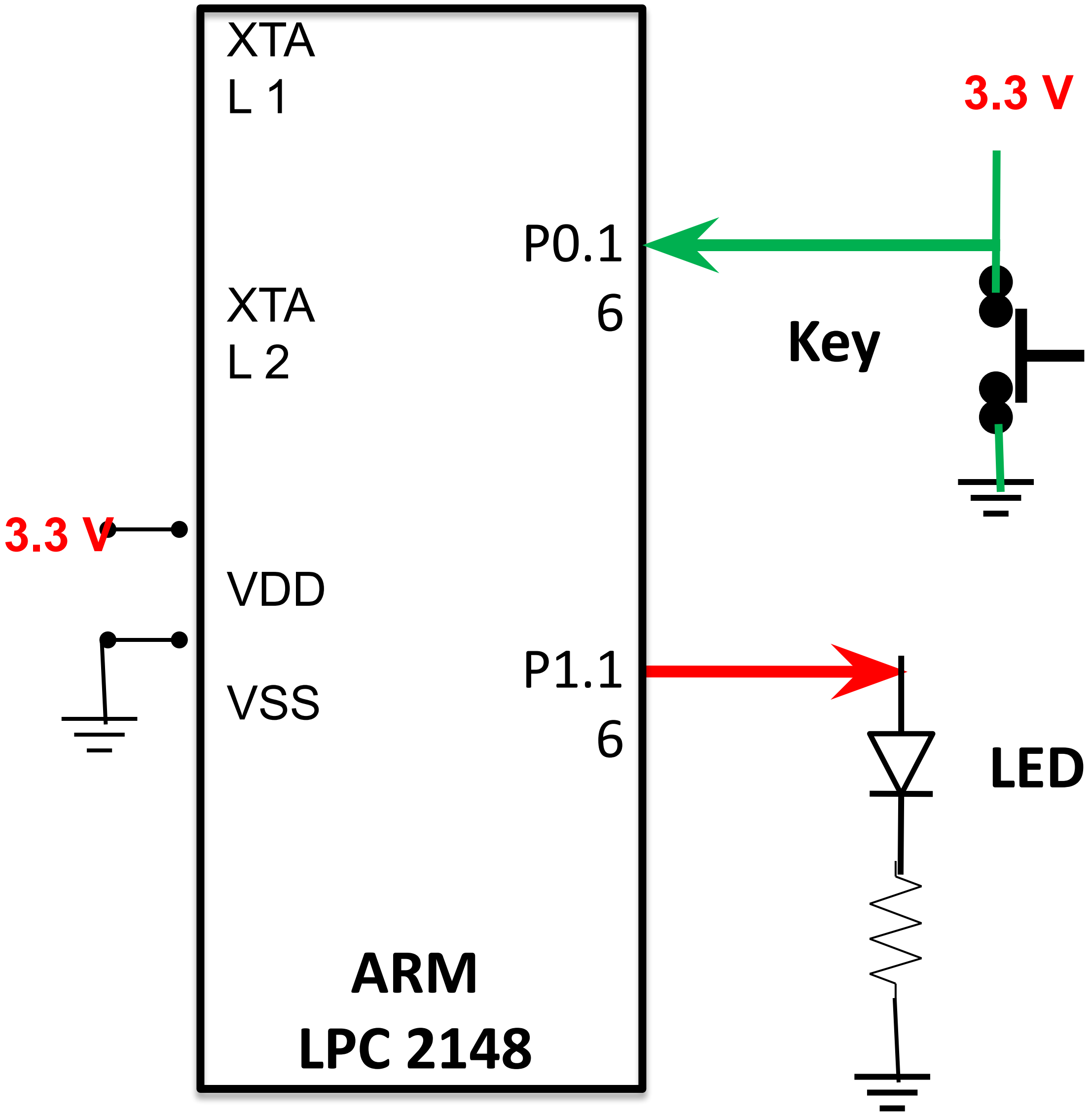
Q. Write program to turn ON LED's if Switch is pressed, 8 LED are connected to
P0.0 to P0.7 and Switch is connected to P1.16

Simulation using Proteus SW



31-Oct-25

Q. Interfacing 8 LED



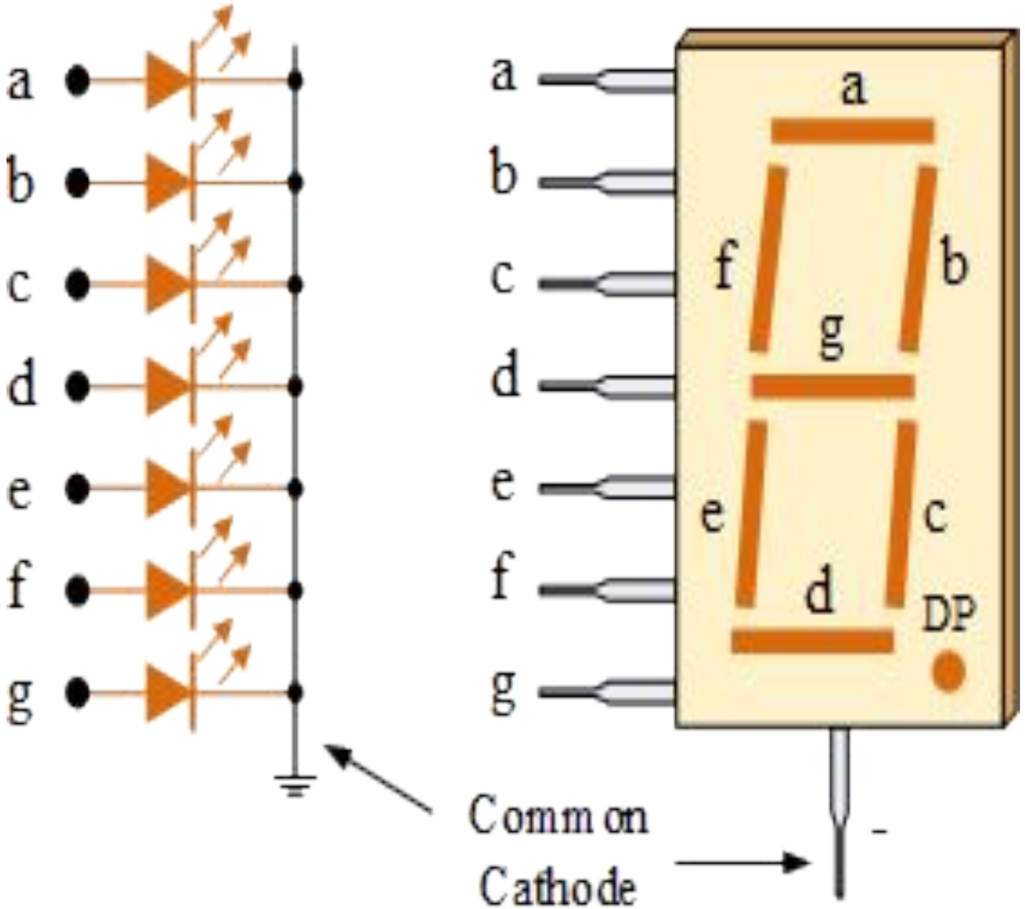
Interfacing 7 segment Display with LPC2148

7 Segment Display

Common Cathode

Data = 0, then LED is OFF

Data = 1, then LED is ON

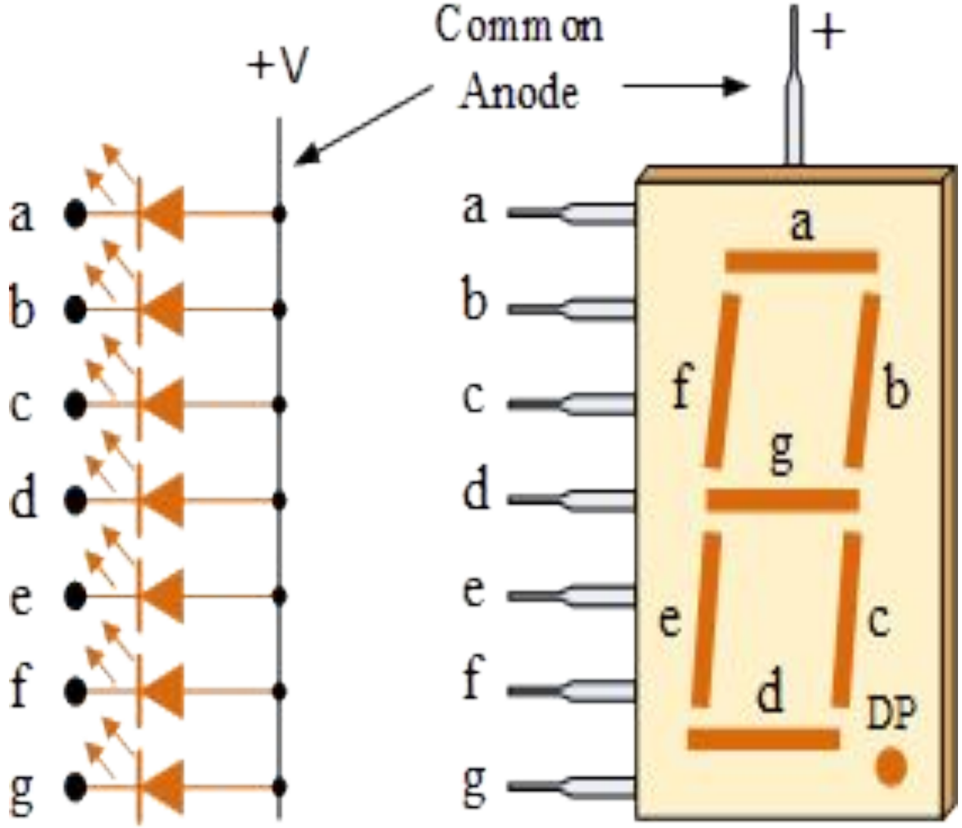


7 Segment Display

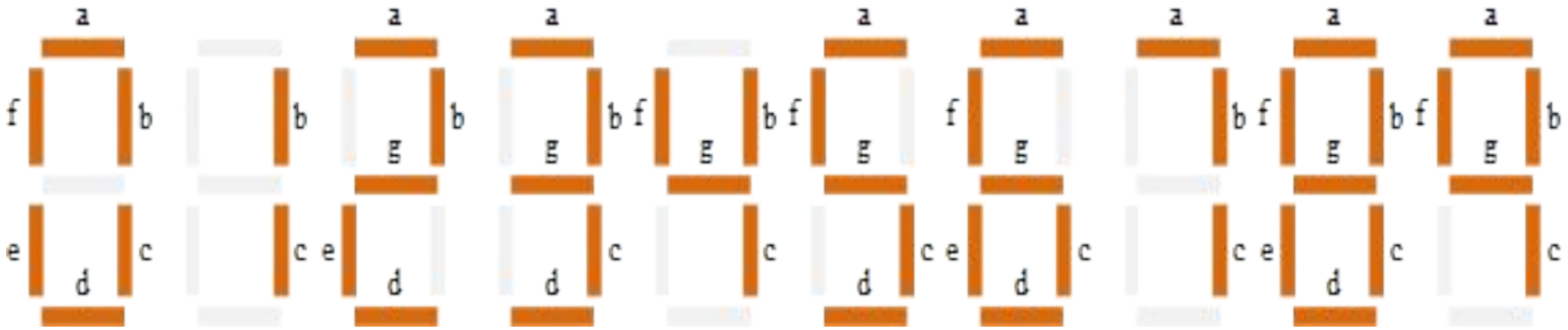
Common Anode

Data = 1,
then LED is
OFF

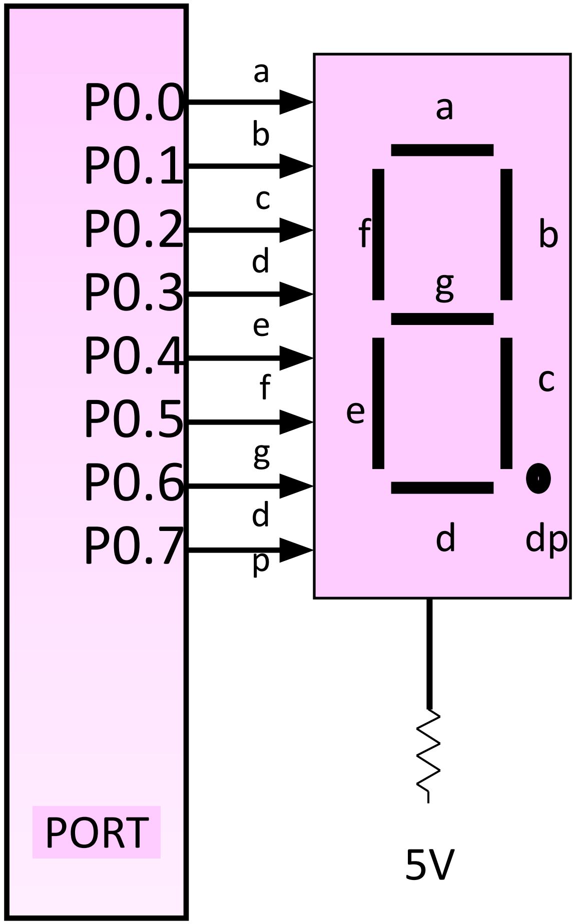
Data = 0,
then LED is
ON



7 Segment Display for Digits to be displayed

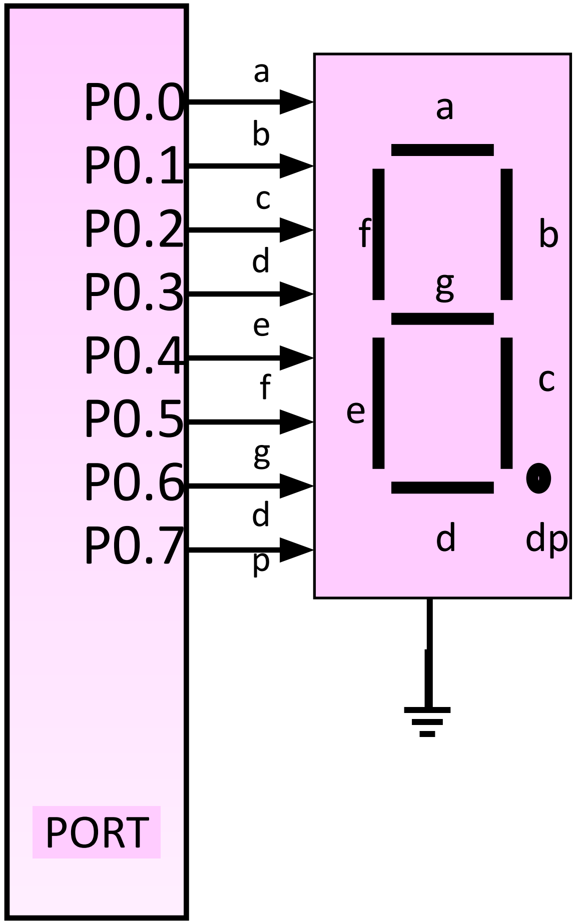


7 Segment Display Code for Common Anode



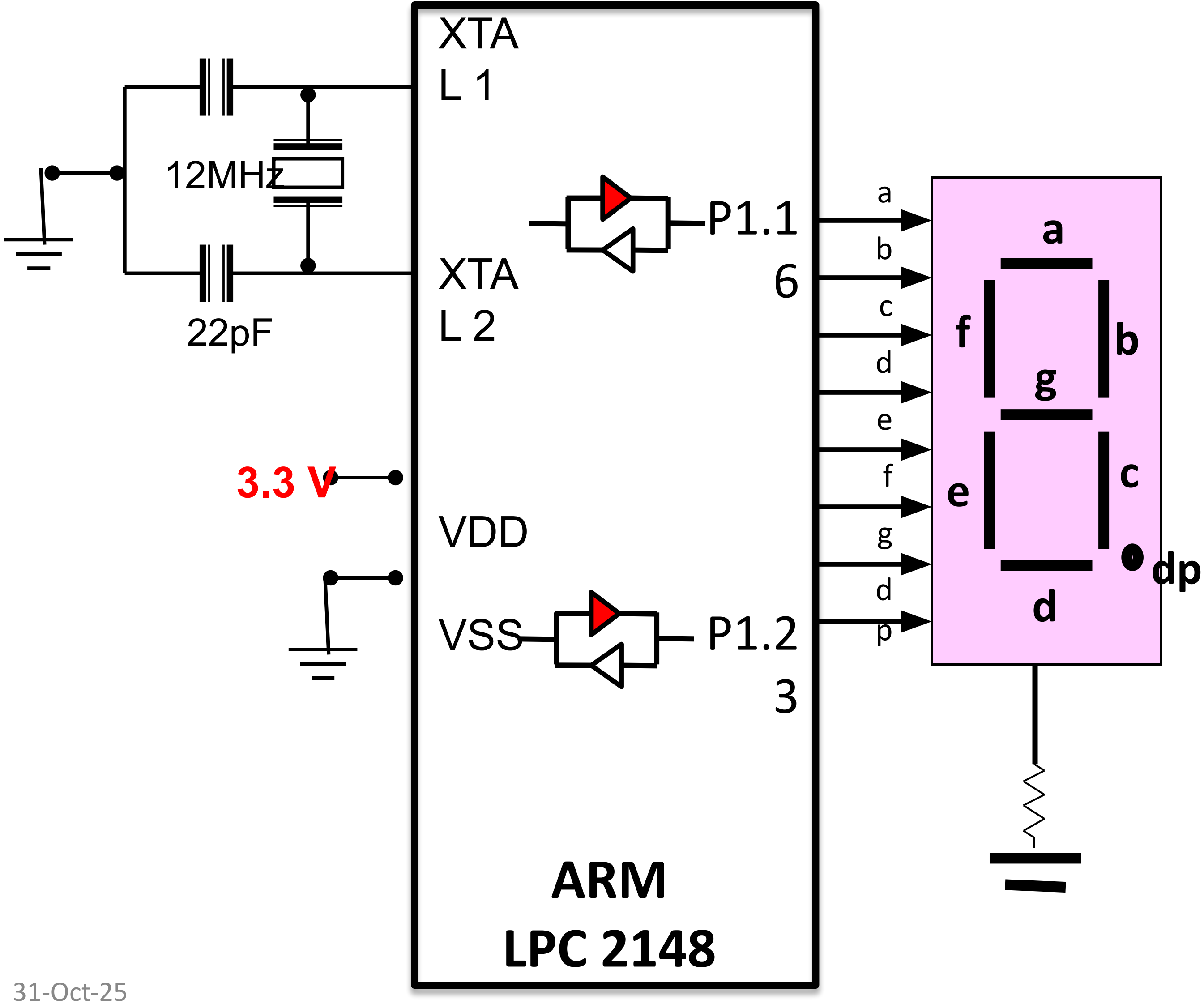
	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	Hex
DIGIT	dp	g	f	e	d	c	b	a	
0	1	1	0	0	0	0	0	0	COH
1	1	1	1	1	1	0	0	1	F9H
2	1	0	1	0	0	1	0	0	A4H
3	1	0	1	1	0	0	0	0	B0H
4	1	0	0	1	1	0	0	1	99H
5	1	0	0	1	0	0	1	0	92H
6	1	0	0	0	0	0	1	0	82H
7	1	1	1	1	1	0	0	0	F8H
8	1	0	0	0	0	0	0	0	80H
9	1	0	0	1	0	0	0	0	90H

7 Segment Display Code for Common Cathode



	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	Hex
DIGIT	dp	g	f	e	d	c	b	a	
0	0	0	1	1	1	1	1	1	**3FH**
1	0	0	0	0	0	1	1	0	**06H**
2	0	1	0	1	1	0	1	1	**5BH**
3	0	1	0	0	1	1	1	1	**4FH**
4	0	1	1	0	0	1	1	0	**66H**
5	0	1	1	0	1	1	0	1	**6DH**
6	0	1	1	1	1	1	0	1	**7DH**
7	0	0	0	0	0	1	1	1	**07H**
8	0	1	1	1	1	1	1	1	**7FH**
9	0	1	1	0	1	1	1	1	**6FH**

Q. Draw interfacing for Common Cathode 7 segment display with LPC2148 and write C language program to display 5. Seven segment display is connected to P1.23 to P1.16



Common Cathode 7 segment display: To display “5” hex code is 6DH

	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	Hex
DIGIT	dp	g	f	e	d	c	b	a	
0	0	0	1	1	1	1	1	1	**3FH**
1	0	0	0	0	0	1	1	0	**06H**
2	0	1	0	1	1	0	1	1	**5BH**
3	0	1	0	0	1	1	1	1	**4FH**
4	0	1	1	0	0	1	1	0	**66H**
5	0	1	1	0	1	1	0	1	**6DH**
6	0	1	1	1	1	1	0	1	**7DH**
7	0	0	0	0	0	1	1	1	**07H**
8	0	1	1	1	1	1	1	1	**7FH**
9	0	1	1	0	1	1	1	1	**6FH**

- **Define GPIO pins P1.23 to P1.16 as output pins**
- **Send hex code 6DH to Port 1 pins (P1.23 to P1.16) to display “5”**
- **32 bit = 0x006D0000**

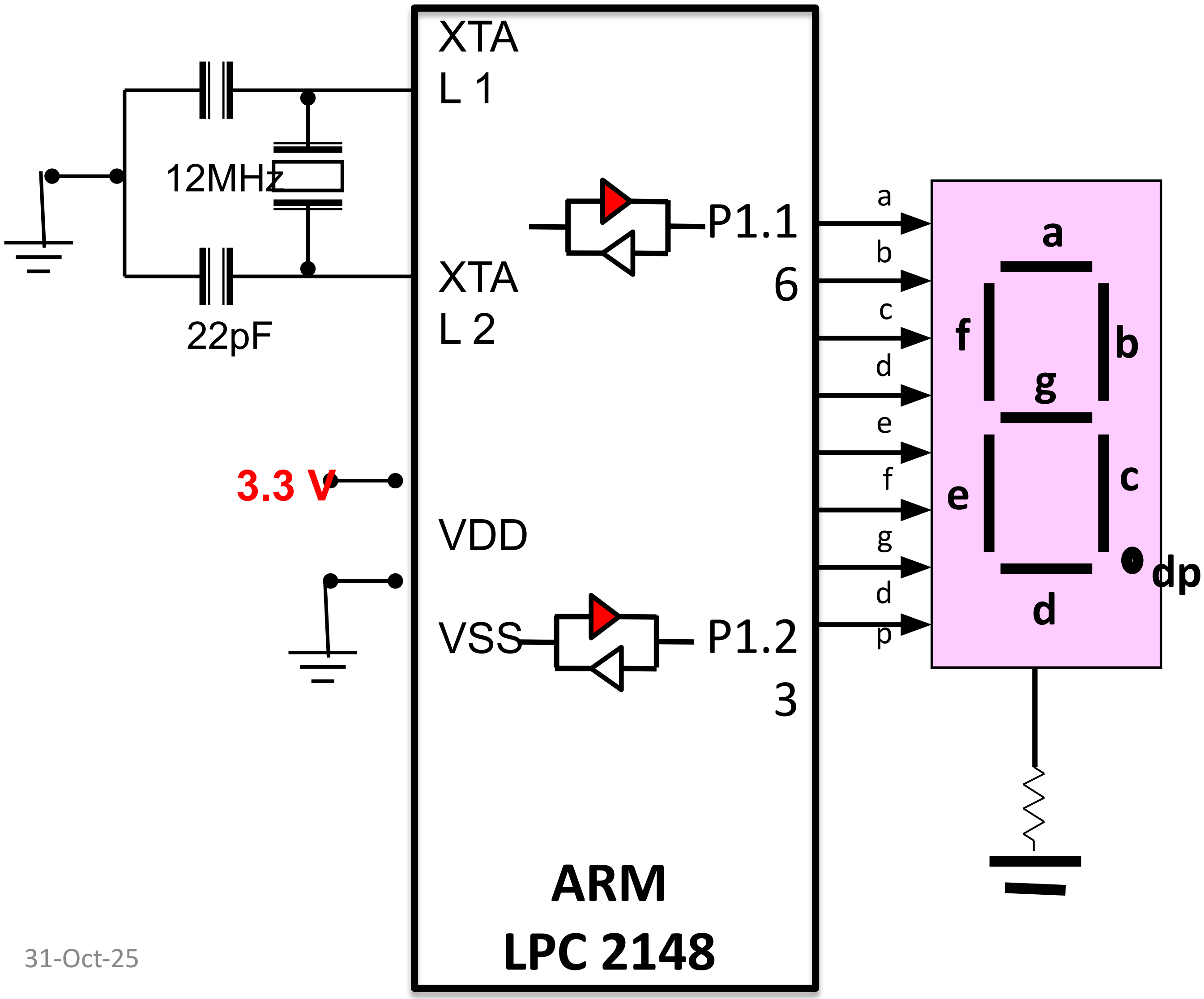
Program To display “5”

```
#include<lpc214x.h>
void main()
{
   IO1DIR =0X00FF0000;            //P1.23 to P1.16 is now acting as a
output pin
   IO1SET =0X006D0000;           // a= 1, b=0, c=1, d=1, e=0, f=1, g=1
}
```

Program To display “5” then “6”

```
#include<lpc214x.h>
void main()
{
   IO1DIR =0X00FF0000;              //P1.23 to P1.16 is now acting as a
output pin
   IO1SET =0X006D0000;             // a= 1, b=0, c=1, d=1, e=0, f=1, g=1
   IO1CLR =0X006D0000;             // a= 0, b=0, c=0, d=0, e=0, f=0, g=0
   IO1SET =0X007D0000;             // a= 1, b=0, c=1, d=1, e=1, f=1, g=1
}
```

Q. Draw interfacing for Common Cathode 7 segment display with LPC2148 and write C language program to display 0 and then 1 after 3 sec delay. Seven segment display is connected to P1.23 to P1.16



Common Cathode 7 segment display: To display “5” and “6”

	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	Hex
DIGIT	dp	g	f	e	d	c	b	a	
0	0	0	1	1	1	1	1	1	**3FH**
1	0	0	0	0	0	1	1	0	**06H**
2	0	1	0	1	1	0	1	1	**5BH**
3	0	1	0	0	1	1	1	1	**4FH**
4	0	1	1	0	0	1	1	0	**66H**
5	0	1	1	0	1	1	0	1	**6DH**
6	0	1	1	1	1	1	0	1	**7DH**
7	0	0	0	0	0	1	1	1	**07H**
8	0	1	1	1	1	1	1	1	**7FH**
9	0	1	1	0	1	1	1	1	**6FH**

- Define GPIO pins P1.23 to P1.16 as output pins
- Send hex code 3FH to Port 1 pins (P1.23 to P1.16) to display “0” 32 bit = 0x003F0000
- Call Delay of 3 sec
- Send hex code 06H to Port 1 pins (P1.23 to P1.16) to display “1” 32 bit = 0x00060000

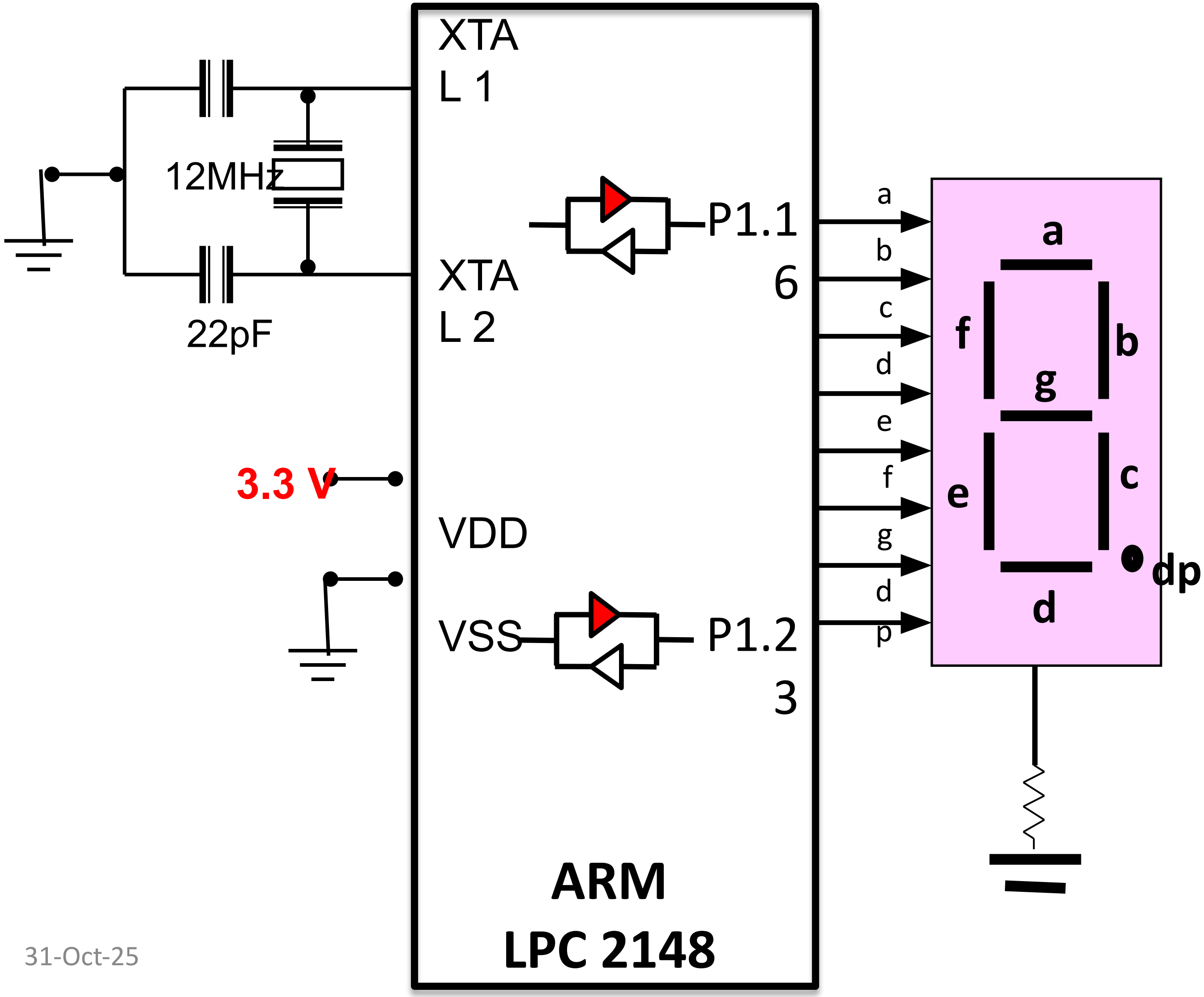
Program

```
#include<lpc214x.h>
void delay(int );
int i;
unsigned int a[]={0x003f0000,0x00060000};
int main()
{
   IO1DIR=IO1DIR|0x00ff0000;
   while(1)
   {
      for(i=0;i<=1;i++)
      {
         IO1SET=IO1SET|a[i];
         delay(300);
         IO1CLR=IO1CLR|a[i];
      }
   }
}
```

Program

```
void delay(int k)
{
    int i,j;
    for(i=0;i<k;i++)
    for(j=0;j<=1000;j++);
}
```

Q. Draw interfacing for Common Cathode 7 segment display with LPC2148 and write C language program to display 0 to 9 with delay of 1 sec. Seven segment display is connected to P1.23 to P1.16



Common Cathode 7 segment display: To display “5” and “6”

	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	Hex
DIGIT	dp	g	f	e	d	c	b	a	
0	0	0	1	1	1	1	1	1	3FH
1	0	0	0	0	0	1	1	0	06H
2	0	1	0	1	1	0	1	1	5BH
3	0	1	0	0	1	1	1	1	4FH
4	0	1	1	0	0	1	1	0	66H
5	0	1	1	0	1	1	0	1	6DH
6	0	1	1	1	1	1	0	1	7DH
7	0	0	0	0	0	1	1	1	07H
8	0	1	1	1	1	1	1	1	7FH
9	0	1	1	0	1	1	1	1	6FH

- Define GPIO pins P1.23 to P1.16 as output pins
- Send hex code 3FH to Port 1 pins (P1.23 to P1.16) to display “0” 32 bit = 0x003F0000
- Call Delay of 3 sec
- Send hex code 06H to Port 1 pins (P1.23 to P1.16) to display “1” 32 bit = 0x00060000

Program

```
#include<lpc214x.h>
void delay(int );
int i;
unsigned int a[]={0x003f0000, 0x00060000, 0x005B0000,
0x004F0000, 0x00660000,
0x006D0000,0x007D0000,0x00070000,0x007F0000,0x006F0000};
int main()
{
   IO1DIR=IO1DIR|0x00ff0000;
   while(1)
   {
      for(i=0;i<=9;i++)
      {
         IO1SET=IO1SET|a[i];
         delay(300);
         IO1CLR=IO1CLR|a[i];
      }
   }
}
```

Program

```
void delay(int k)
{
    int i,j;
    for(i=0;i<k;i++)
    for(j=0;j<=1000;j++);
}
```