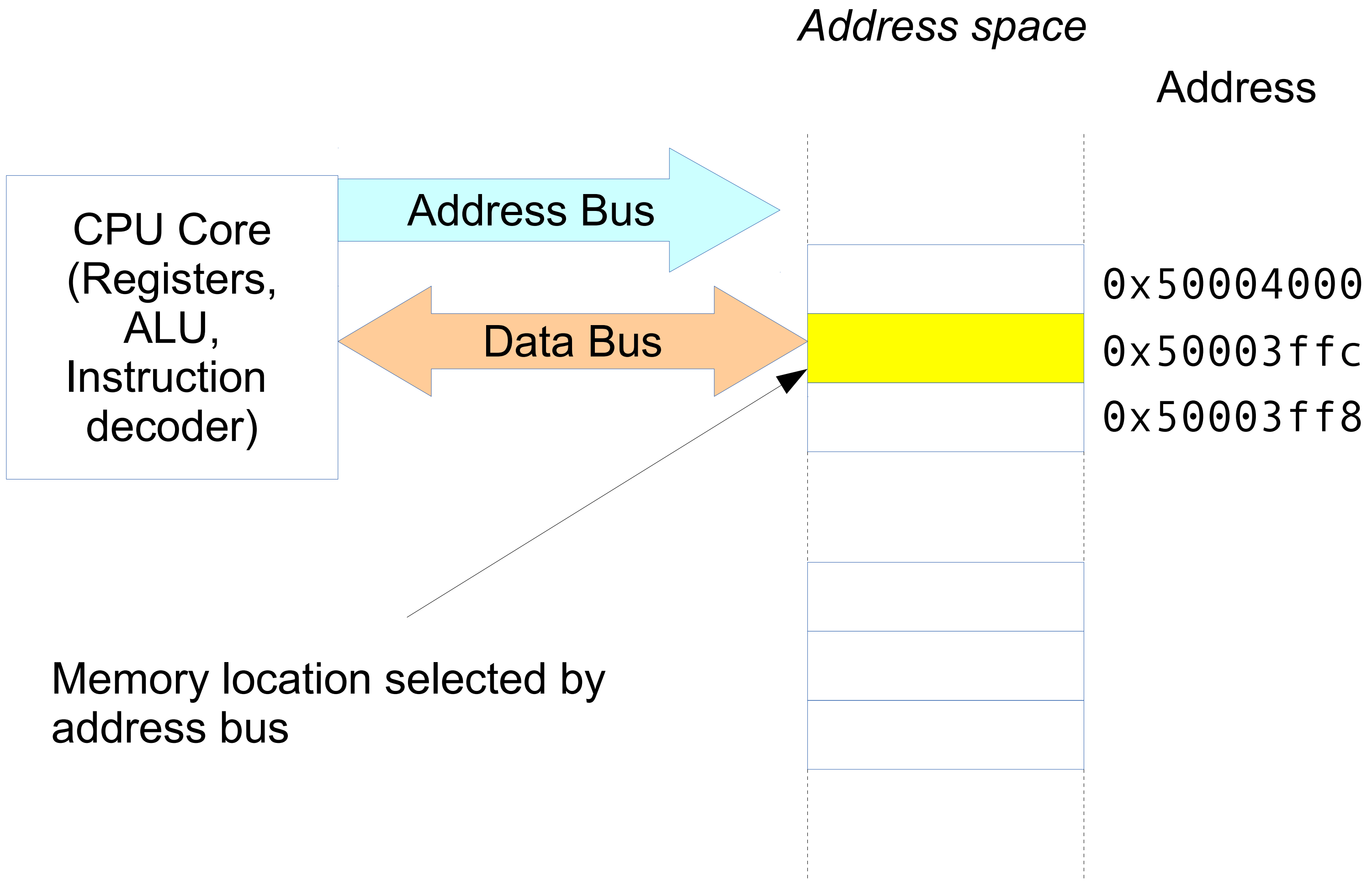
Parallel Input/Output

Parallel Input/Output

Called General Pupose Input Ouput
Or
GPIO PORTS
in LPC1114 documents

Parallel Input/Output

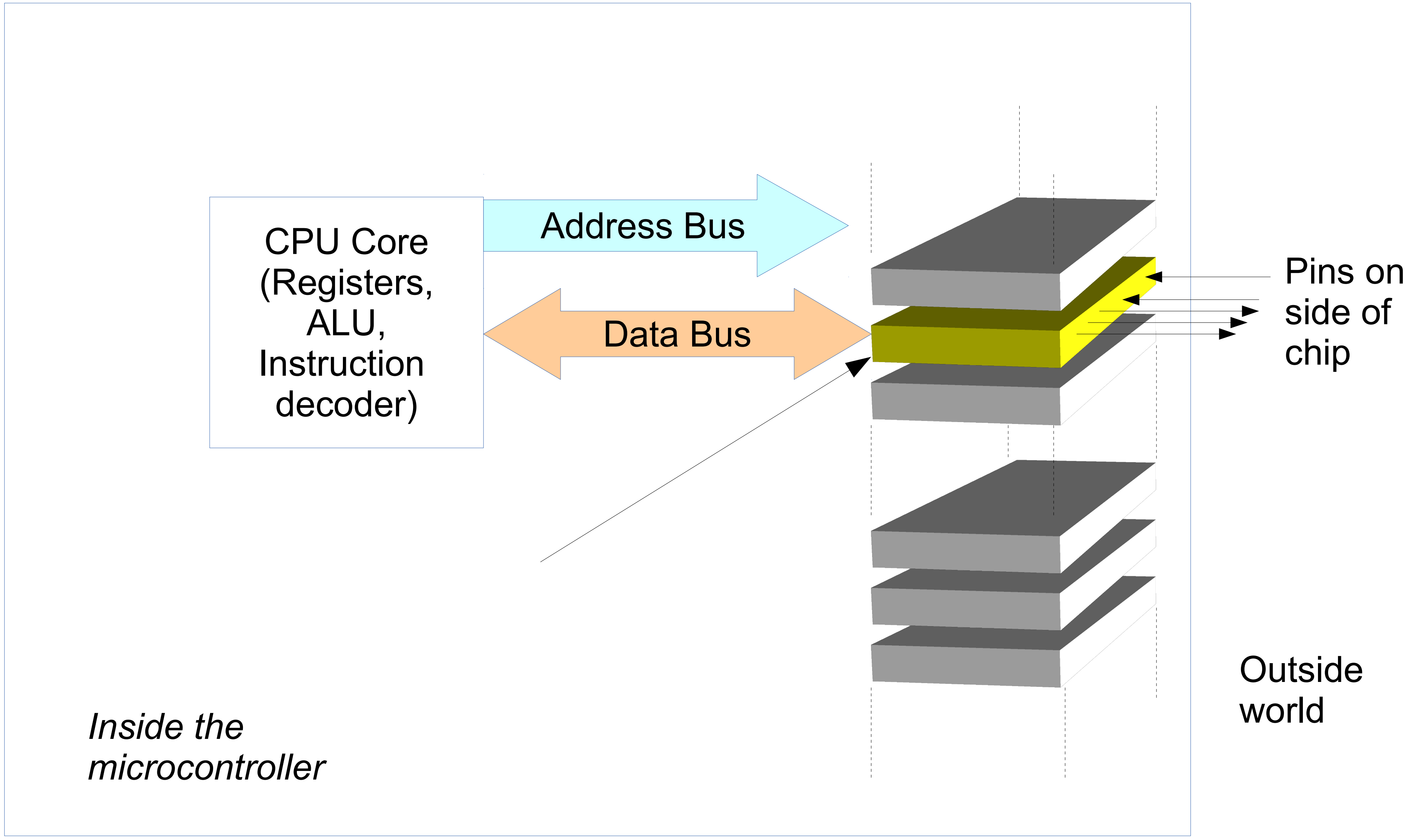

Address space
Address
CPU Core (Registers, ALU, Instruction decoder)
Address Bus
Data Bus
0x50004000
0x50003ffc
0x50003ff8
Memory location selected by address bus

CPU Addressing a memory location

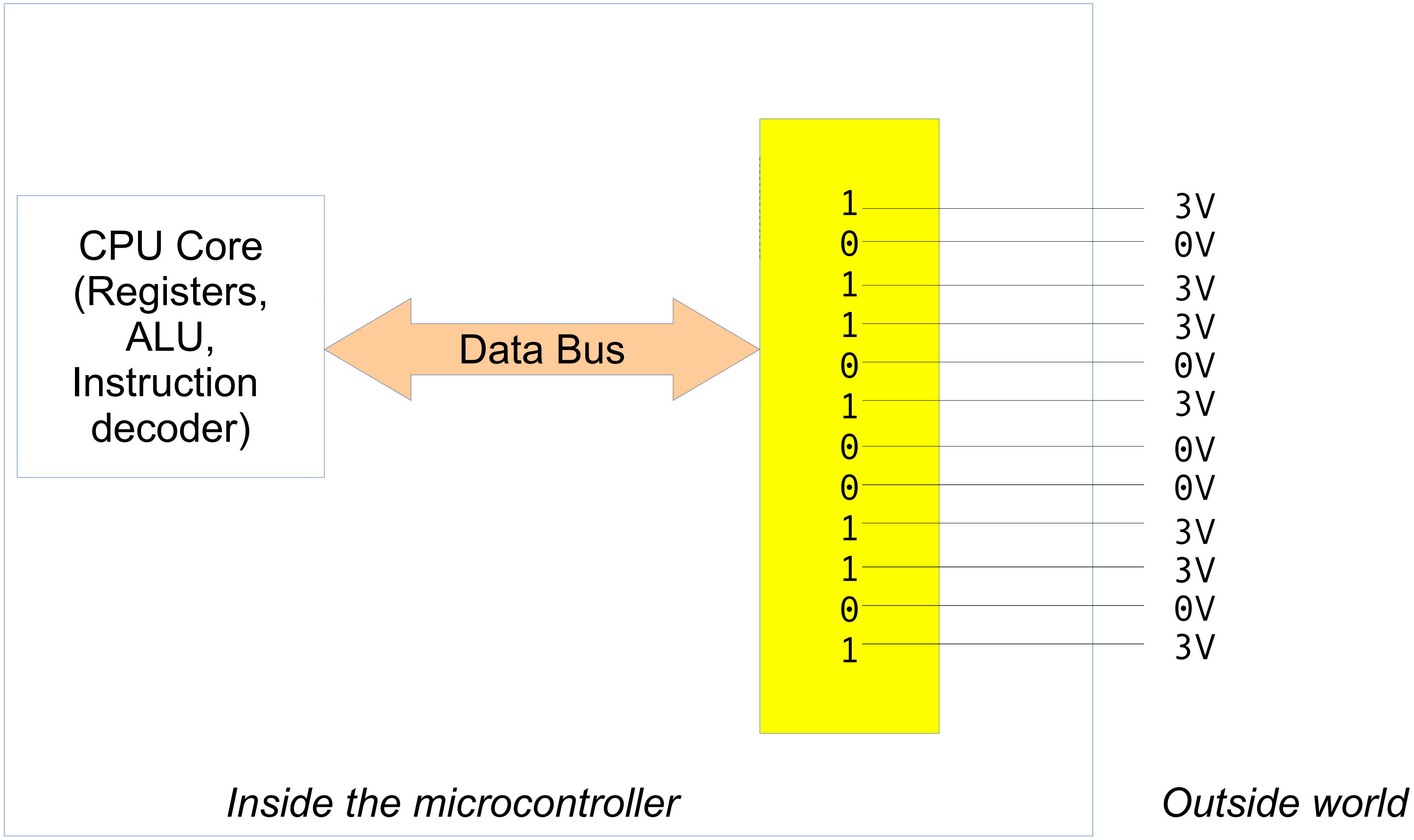
Parallel Input/Output

- Most memory locations only connect to data bus
- Some special memory locations also connect to other things
  - Internal devices such as ADC's, Timer etc.
  - External devices via pins on the chip : **GPIO**

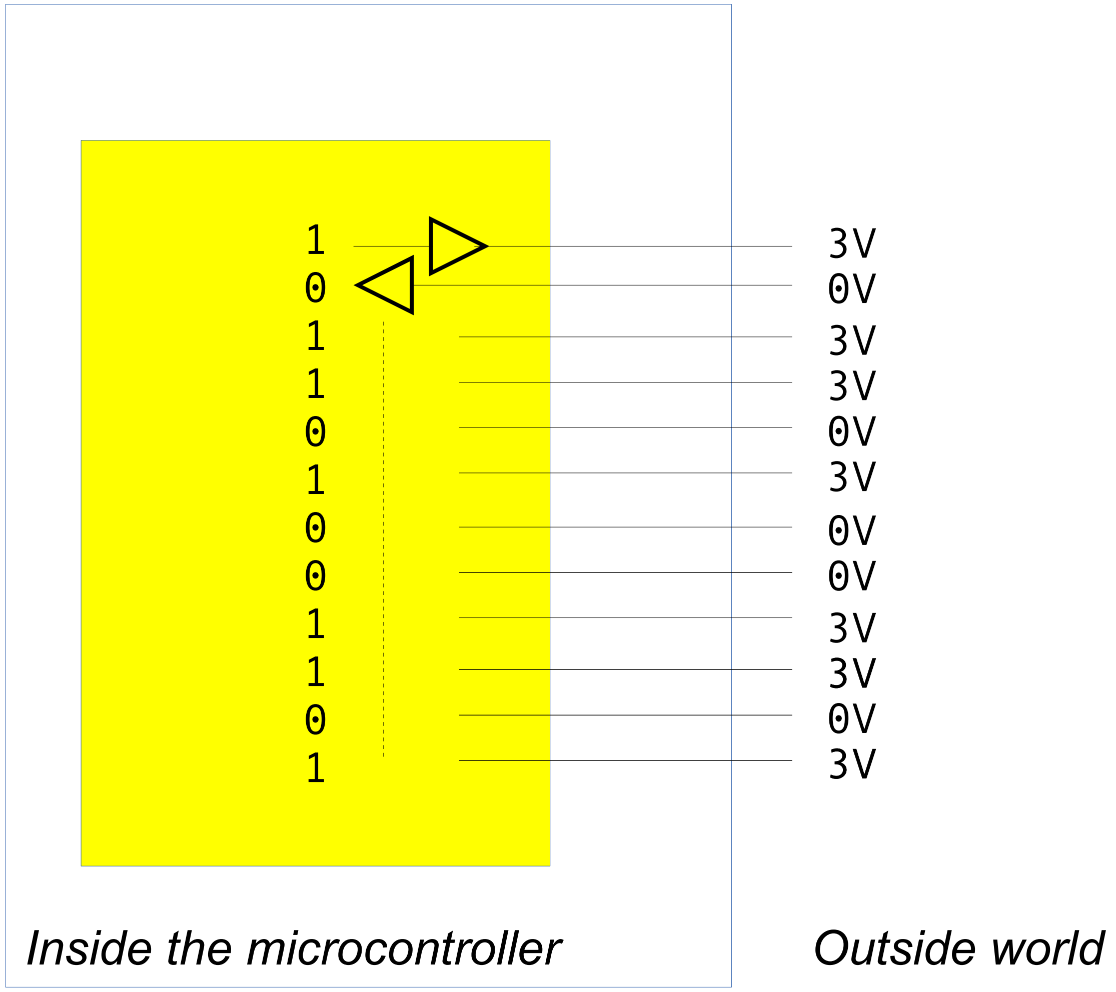
Parallel Input/Output


CPU Core (Registers, ALU, Instruction decoder)
Address Bus
Data Bus
Pins on side of chip
Outside world
Inside the microcontroller

Parallel Input/Output


CPU Core (Registers, ALU, Instruction decoder)
Data Bus
1 3V
0 0V
1 3V
1 3V
0 0V
1 3V
0 0V
0 0V
1 3V
1 3V
0 0V
1 3V
Inside the microcontroller
Outside world

Parallel Input/Output



Some pins can be outputs, some inputs

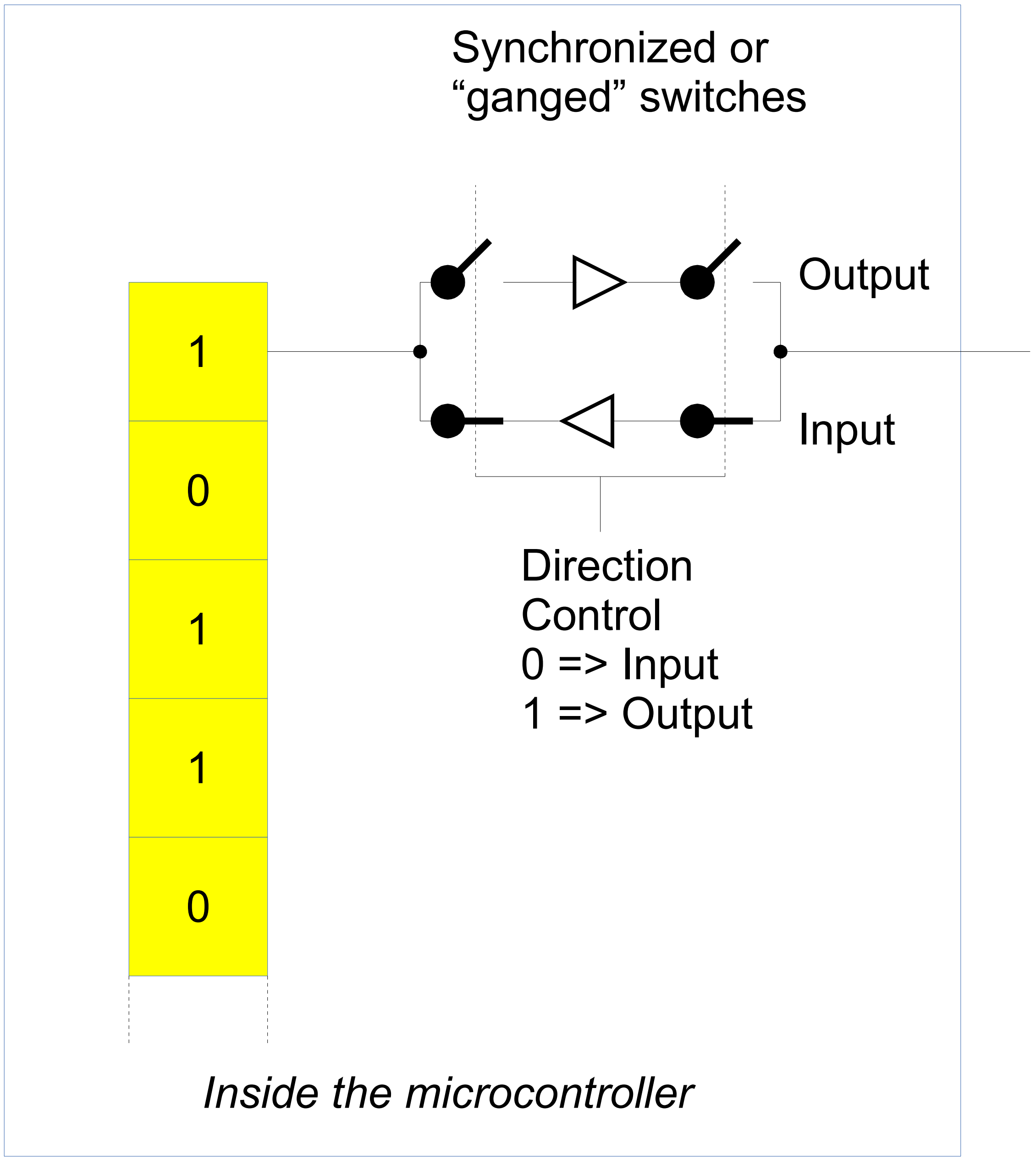
Parallel Input/Output

- Electronics imposes constraints
- A pin can be an input
- A pin can be an output
- NOT BOTH AT THE SAME

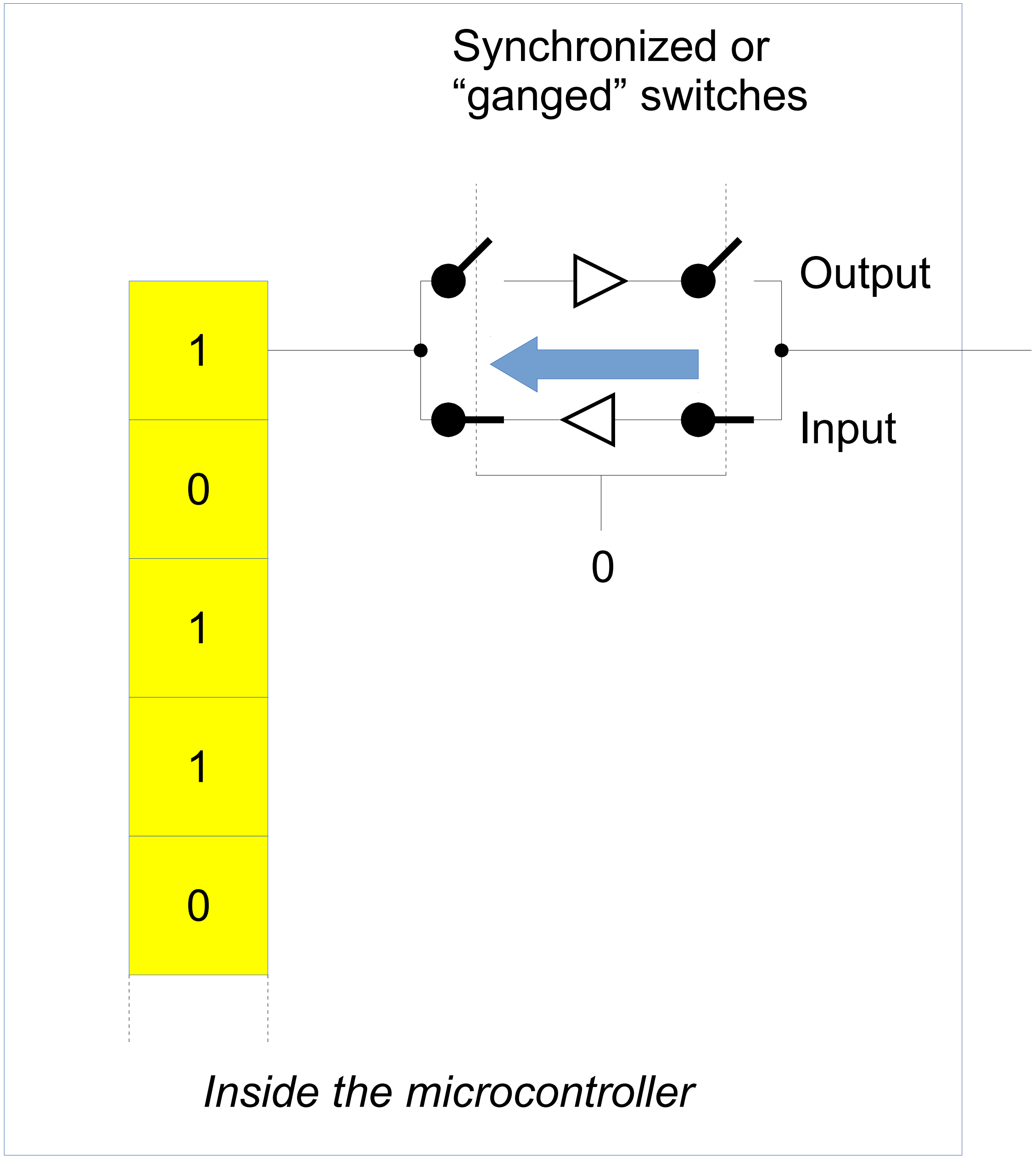
Parallel Input/Output

- How is the direction (input/output) decided?
  - Hardwired by manufacturer
    - Not flexible
  - Controlled by end user software
    - Flexible but requires additional code (and care)

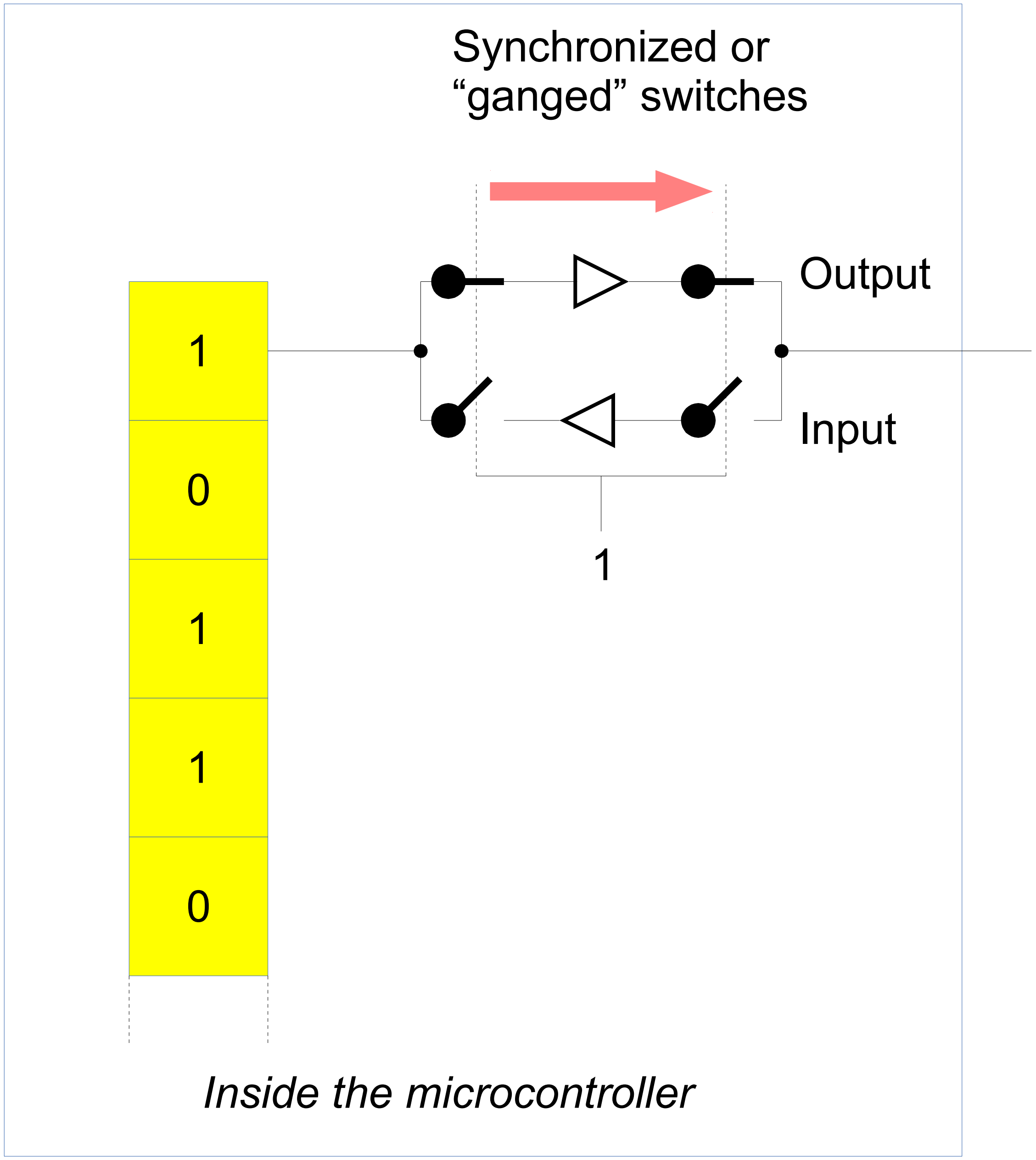
Parallel Input/Output



Parallel Input/Output



Parallel Input/Output



Parallel Input/Output

- Each GPIO port bit is configurable
- A DIRection register is used to control direction
- Each bit in GPIO port register is controlled by corresponding bit in GPIO DIRection register

Parallel Input/Output

- Memory location that sends signals to GPIO bits is called a PORT
- LPC1114 has more than one port
- Ports are numbered 0,1,2, etc.
- e.g.
  - GPIO0
  - GPIO1

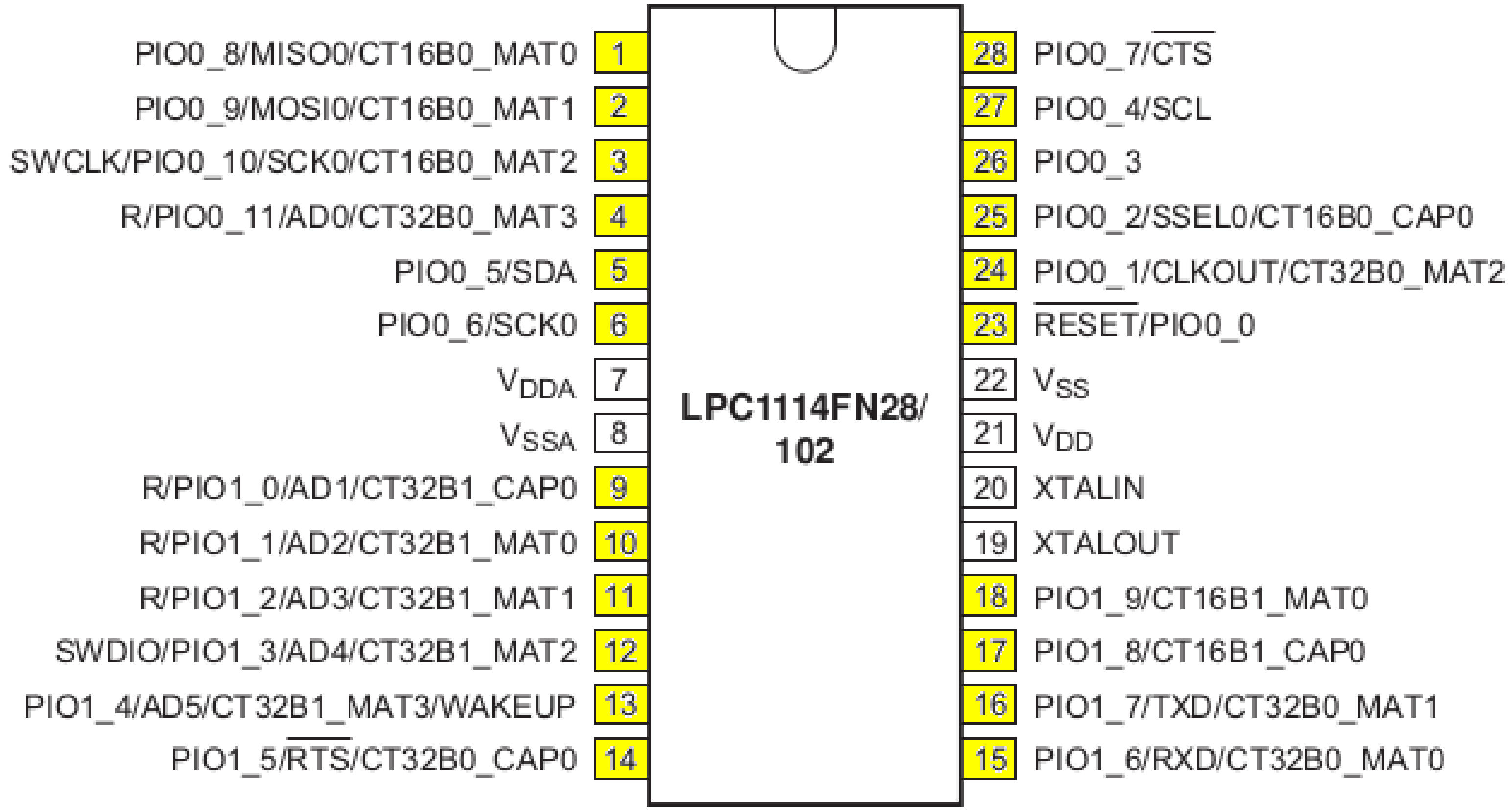
Parallel Input/Output

- Each GPIO port has its own set of control registers and DATA registers
- e.g.
  - GPIO0DATA : holds the bits that drives/read the pins for port 0
  - GPIO0DIR : controls the direction of GPIO0DATA bits

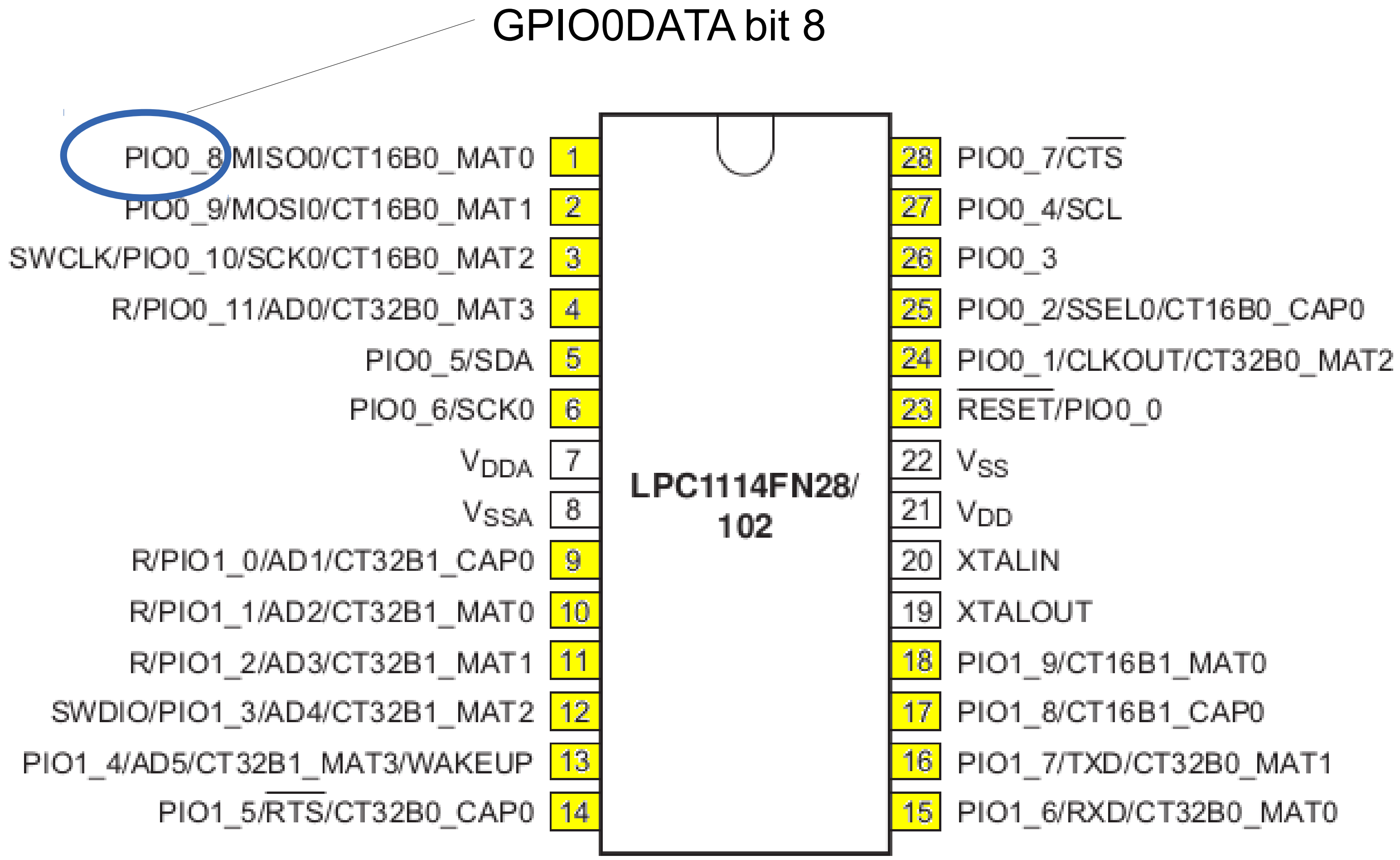
Parallel Input/Output

- Each GPIO port has its own set of control registers and DATA registers
- e.g.
  - GPIO0DATA : holds the bits that drives/read the pins for port 0
  - GPIO0DIR : controls the direction of GPIO0DATA bits

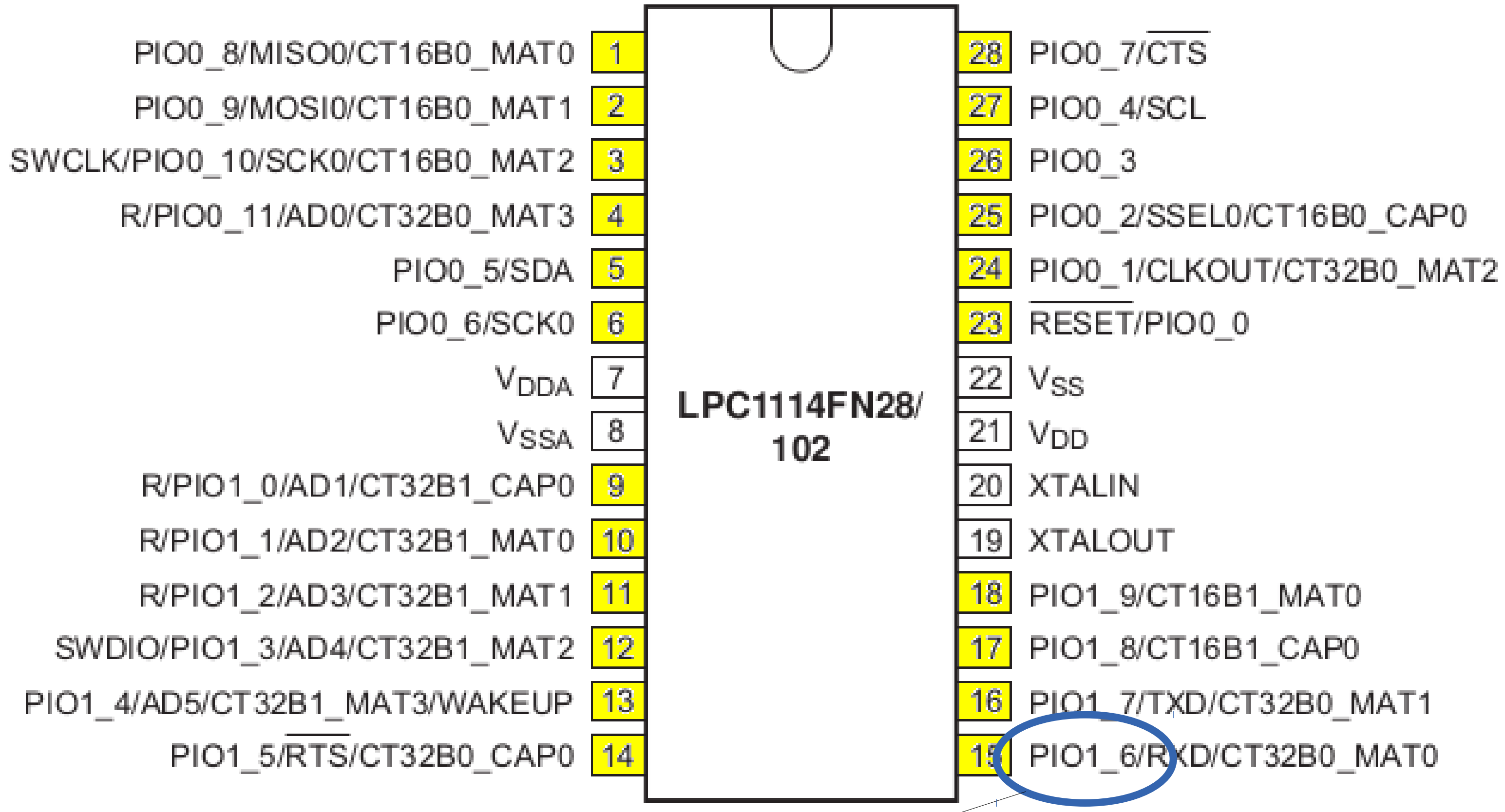
Parallel Input/Output


LPC1114FN28/102
PIO0_8/MISO0/CT16B0_MAT0 1
PIO0_9/MOSI0/CT16B0_MAT1 2
SWCLK/PIO0_10/SCK0/CT16B0_MAT2 3
R/PIO0_11/AD0/CT32B0_MAT3 4
PIO0_5/SDA 5
PIO0_6/SCK0 6
VDDA 7
VSSA 8
R/PIO1_0/AD1/CT32B1_CAP0 9
R/PIO1_1/AD2/CT32B1_MAT0 10
R/PIO1_2/AD3/CT32B1_MAT1 11
SWDIO/PIO1_3/AD4/CT32B1_MAT2 12
PIO1_4/AD5/CT32B1_MAT3/WAKEUP 13
PIO1_5/RTS/CT32B0_CAP0 14
15 PIO1_6/RXD/CT32B0_MAT0
16 PIO1_7/TXD/CT32B0_MAT1
17 PIO1_8/CT16B1_CAP0
18 PIO1_9/CT16B1_MAT0
19 XTALOUT
20 XTALIN
21 VDD
22 VSS
23 RESET/PIO0_0
24 PIO0_1/CLKOUT/CT32B0_MAT2
25 PIO0_2/SSEL0/CT16B0_CAP0
26 PIO0_3
27 PIO0_4/SCL
28 PIO0_7/CTS

Parallel Input/Output

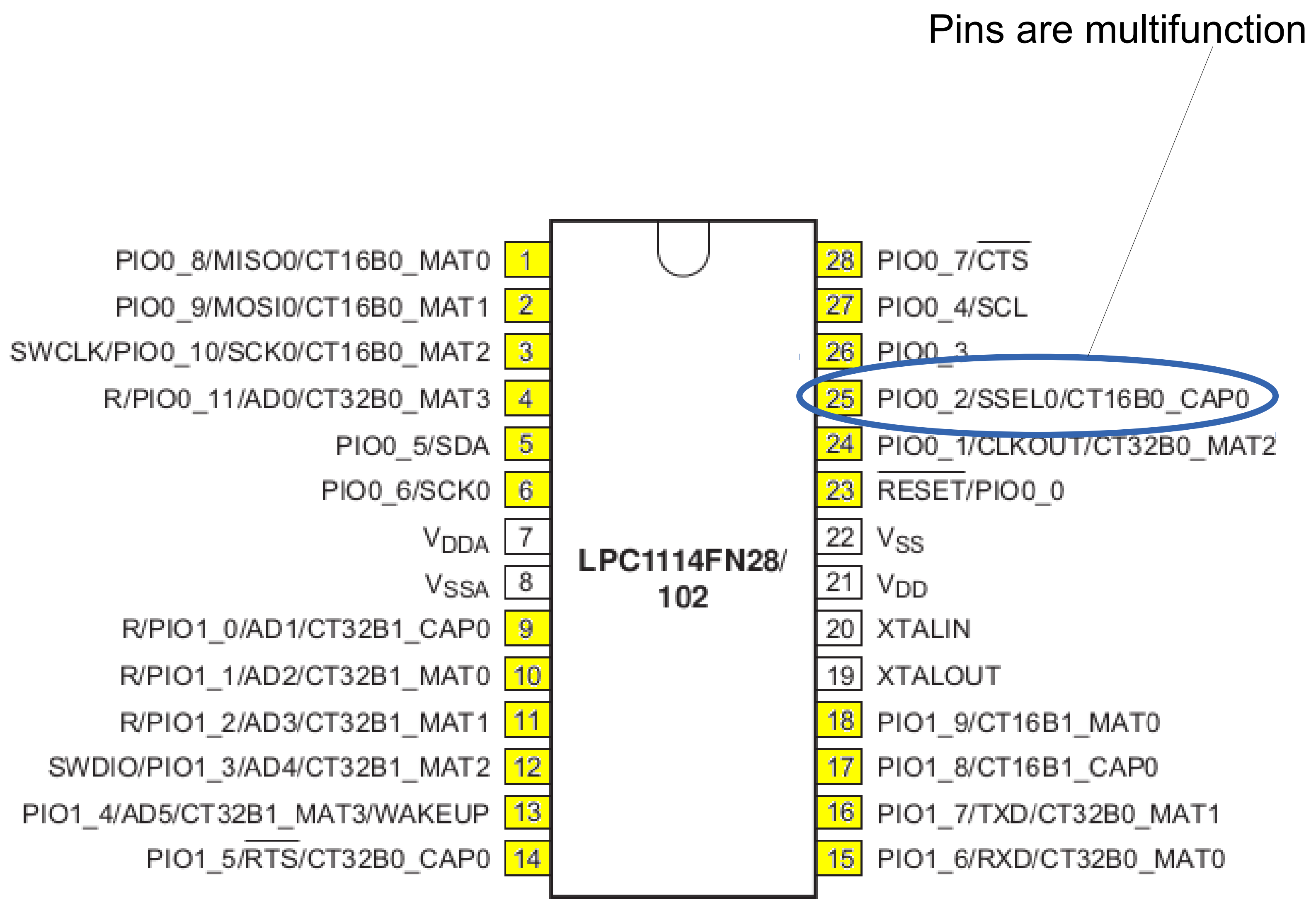

GPIO0DATA bit 8
PIO0_8/MISO0/CT16B0_MAT0 1
PIO0_9/MOSI0/CT16B0_MAT1 2
SWCLK/PIO0_10/SCK0/CT16B0_MAT2 3
R/PIO0_11/AD0/CT32B0_MAT3 4
PIO0_5/SDA 5
PIO0_6/SCK0 6
V_{DDA} 7
V_{SSA} 8
R/PIO1_0/AD1/CT32B1_CAP0 9
R/PIO1_1/AD2/CT32B1_MAT0 10
R/PIO1_2/AD3/CT32B1_MAT1 11
SWDIO/PIO1_3/AD4/CT32B1_MAT2 12
PIO1_4/AD5/CT32B1_MAT3/WAKEUP 13
PIO1_5/$\overline{RTS}$/CT32B0_CAP0 14
LPC1114FN28/102
28 PIO0_7/$\overline{CTS}$
27 PIO0_4/SCL
26 PIO0_3
25 PIO0_2/SSEL0/CT16B0_CAP0
24 PIO0_1/CLKOUT/CT32B0_MAT2
23 $\overline{RESET}$/PIO0_0
22 V_{SS}
21 V_{DD}
20 XTALIN
19 XTALOUT
18 PIO1_9/CT16B1_MAT0
17 PIO1_8/CT16B1_CAP0
16 PIO1_7/TXD/CT32B0_MAT1
15 PIO1_6/RXD/CT32B0_MAT0

Parallel Input/Output

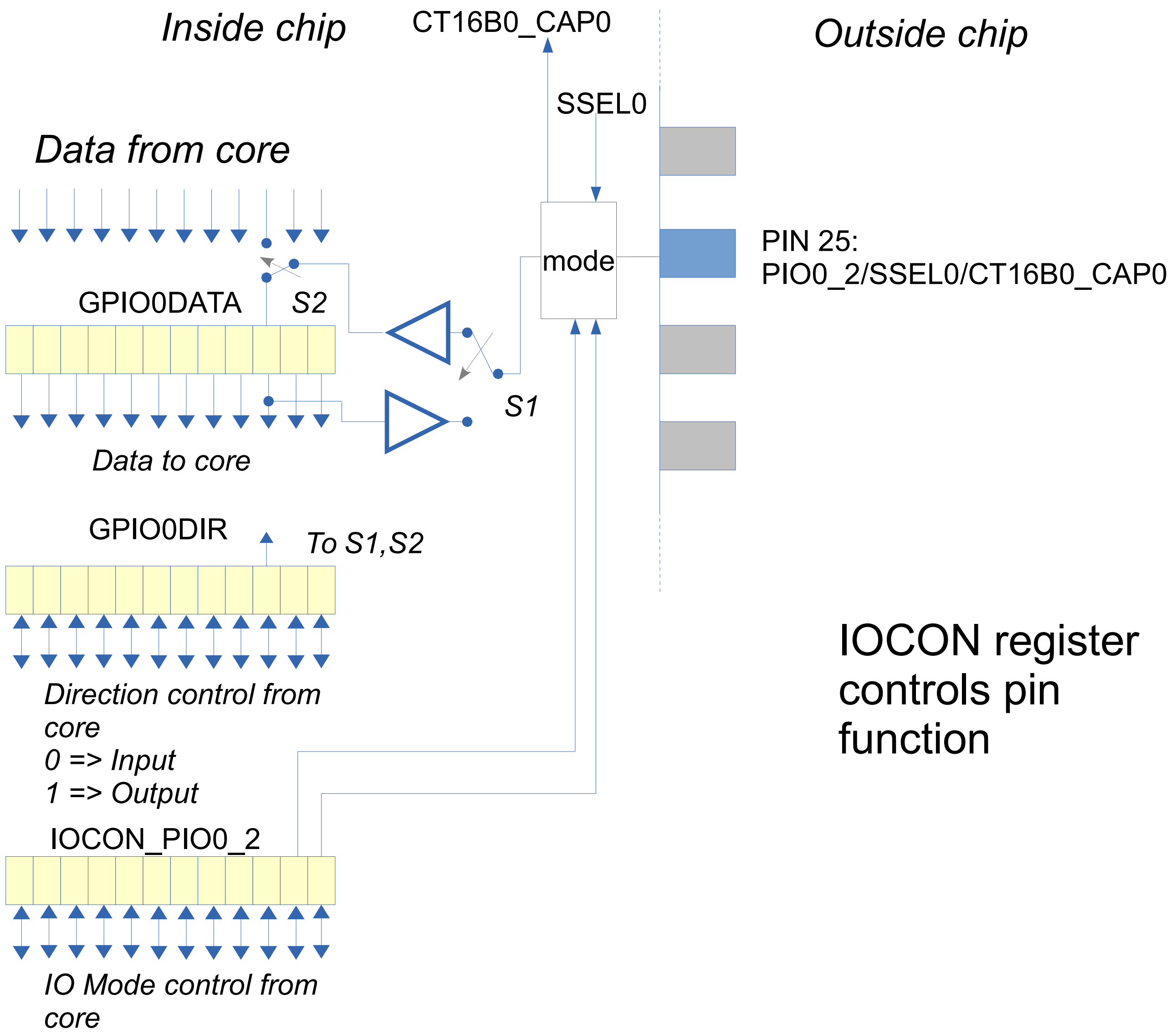

PIO0_8/MISO0/CT16B0_MAT0 1
PIO0_9/MOSI0/CT16B0_MAT1 2
SWCLK/PIO0_10/SCK0/CT16B0_MAT2 3
R/PIO0_11/AD0/CT32B0_MAT3 4
PIO0_5/SDA 5
PIO0_6/SCK0 6
VDDA 7
VSSA 8
R/PIO1_0/AD1/CT32B1_CAP0 9
R/PIO1_1/AD2/CT32B1_MAT0 10
R/PIO1_2/AD3/CT32B1_MAT1 11
SWDIO/PIO1_3/AD4/CT32B1_MAT2 12
PIO1_4/AD5/CT32B1_MAT3/WAKEUP 13
PIO1_5/RTS/CT32B0_CAP0 14
LPC1114FN28/102
28 PIO0_7/CTS
27 PIO0_4/SCL
26 PIO0_3
25 PIO0_2/SSEL0/CT16B0_CAP0
24 PIO0_1/CLKOUT/CT32B0_MAT2
23 RESET/PIO0_0
22 VSS
21 VDD
20 XTALIN
19 XTALOUT
18 PIO1_9/CT16B1_MAT0
17 PIO1_8/CT16B1_CAP0
16 PIO1_7/TXD/CT32B0_MAT1
15 PIO1_6/RXD/CT32B0_MAT0

GPIO1DATA bit 6

Parallel Input/Output


Pins are multifunction
LPC1114FN28/102
1 PIO0_8/MISO0/CT16B0_MAT0
2 PIO0_9/MOSI0/CT16B0_MAT1
3 SWCLK/PIO0_10/SCK0/CT16B0_MAT2
4 R/PIO0_11/AD0/CT32B0_MAT3
5 PIO0_5/SDA
6 PIO0_6/SCK0
7 V_{DDA}
8 V_{SSA}
9 R/PIO1_0/AD1/CT32B1_CAP0
10 R/PIO1_1/AD2/CT32B1_MAT0
11 R/PIO1_2/AD3/CT32B1_MAT1
12 SWDIO/PIO1_3/AD4/CT32B1_MAT2
13 PIO1_4/AD5/CT32B1_MAT3/WAKEUP
14 PIO1_5/$\overline{\text{RTS}}$/CT32B0_CAP0
28 PIO0_7/$\overline{\text{CTS}}$
27 PIO0_4/SCL
26 PIO0_3
25 PIO0_2/SSEL0/CT16B0_CAP0
24 PIO0_1/CLKOUT/CT32B0_MAT2
23 $\overline{\text{RESET}}$/PIO0_0
22 V_{SS}
21 V_{DD}
20 XTALIN
19 XTALOUT
18 PIO1_9/CT16B1_MAT0
17 PIO1_8/CT16B1_CAP0
16 PIO1_7/TXD/CT32B0_MAT1
15 PIO1_6/RXD/CT32B0_MAT0

Parallel Input/Output

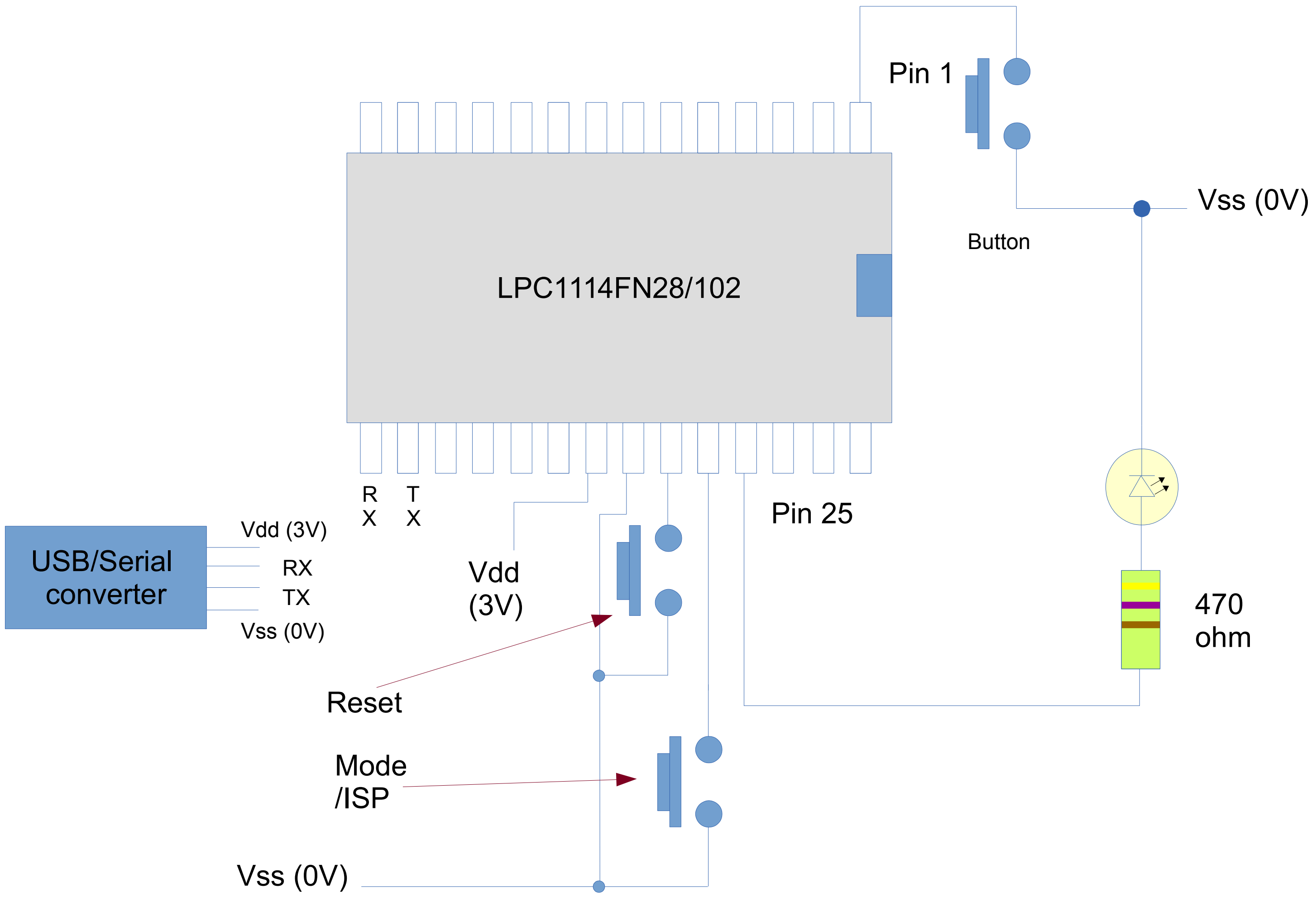


IOCON register controls pin function

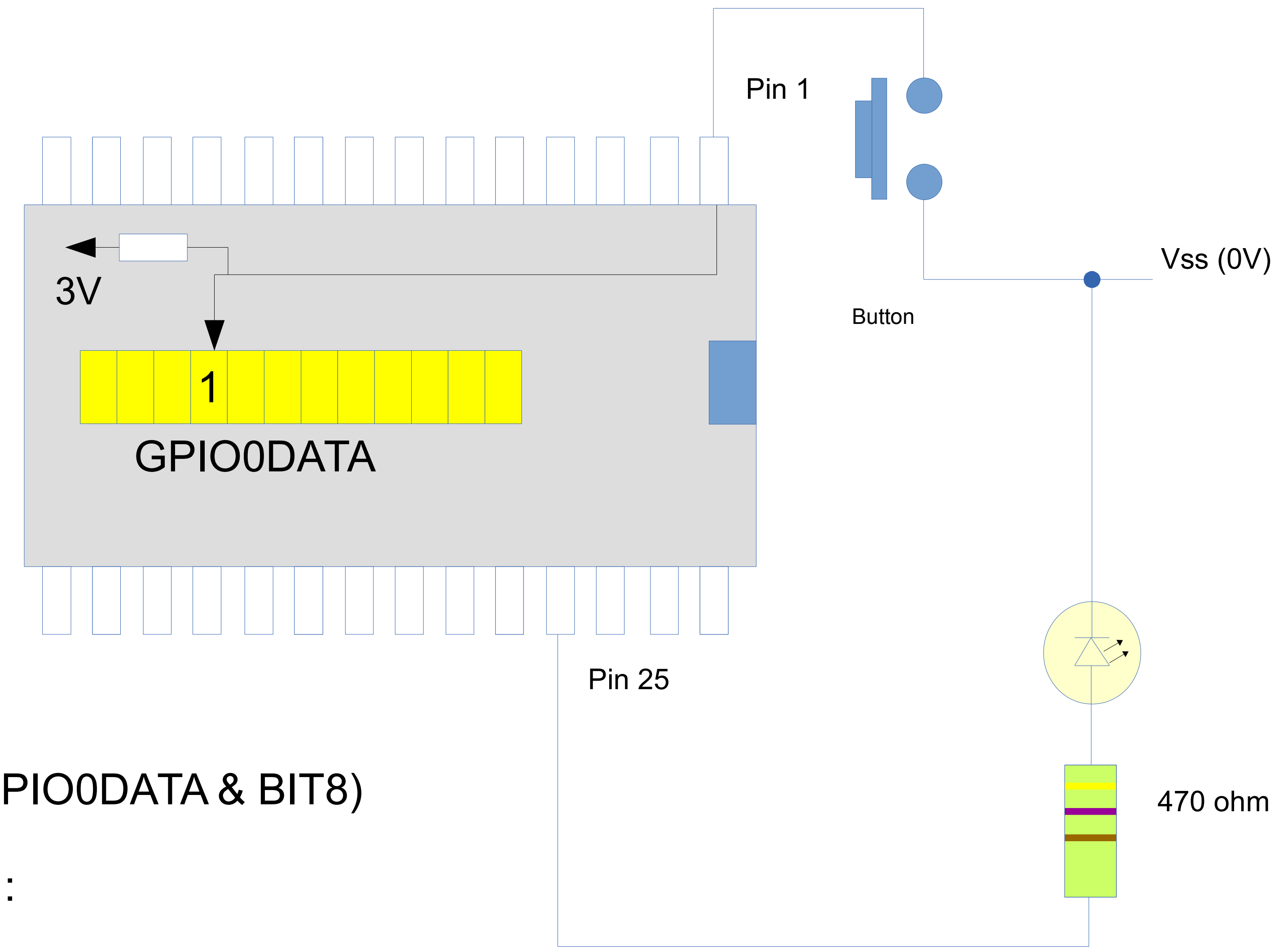
Parallel Input/Output

- Programming GPIO “Pattern”
  - Turn on the GPIO port
  - Set pin function
  - Set pin direction
  - Perform I/O

Parallel Input/Output

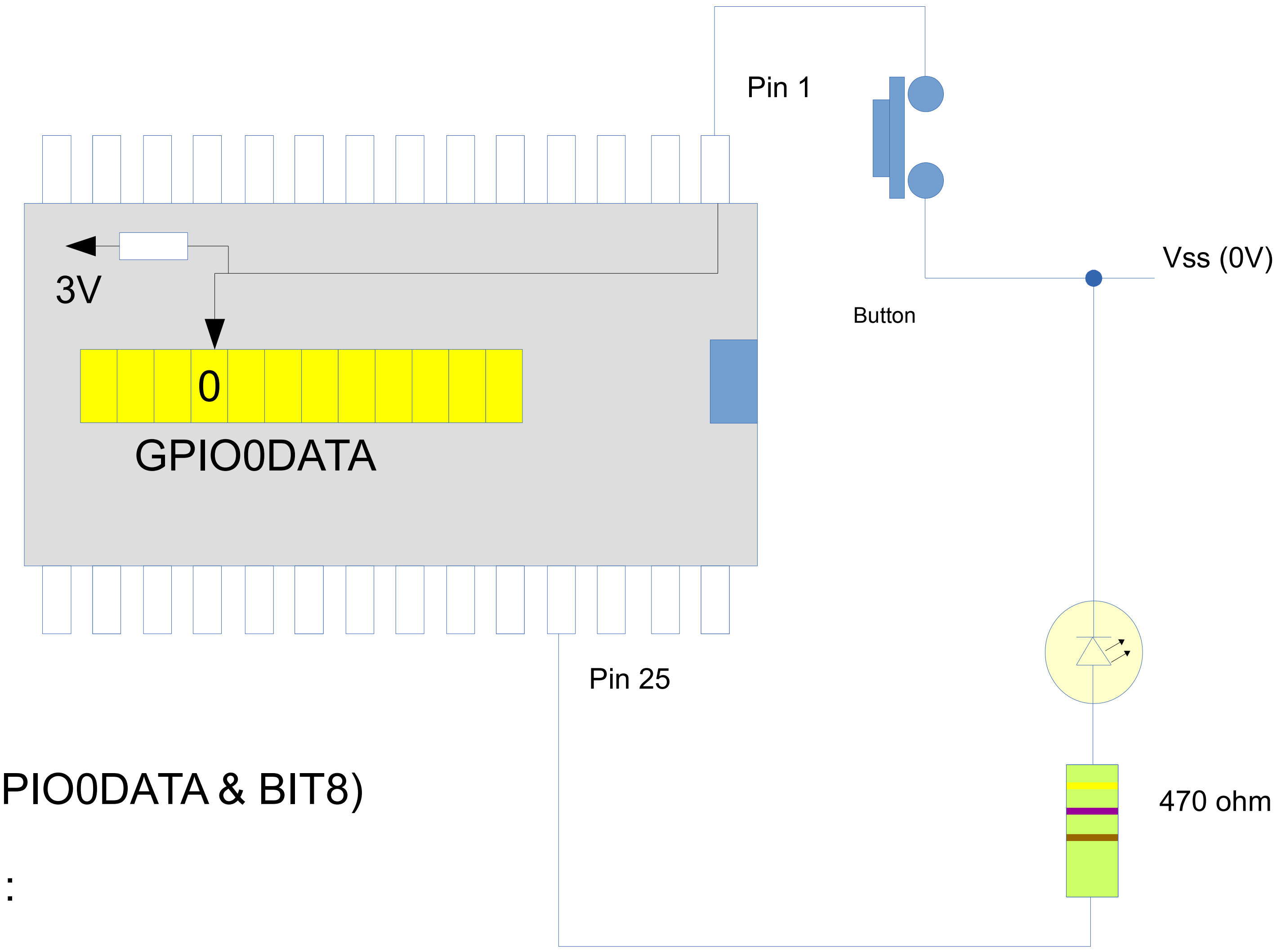


Parallel Input/Output



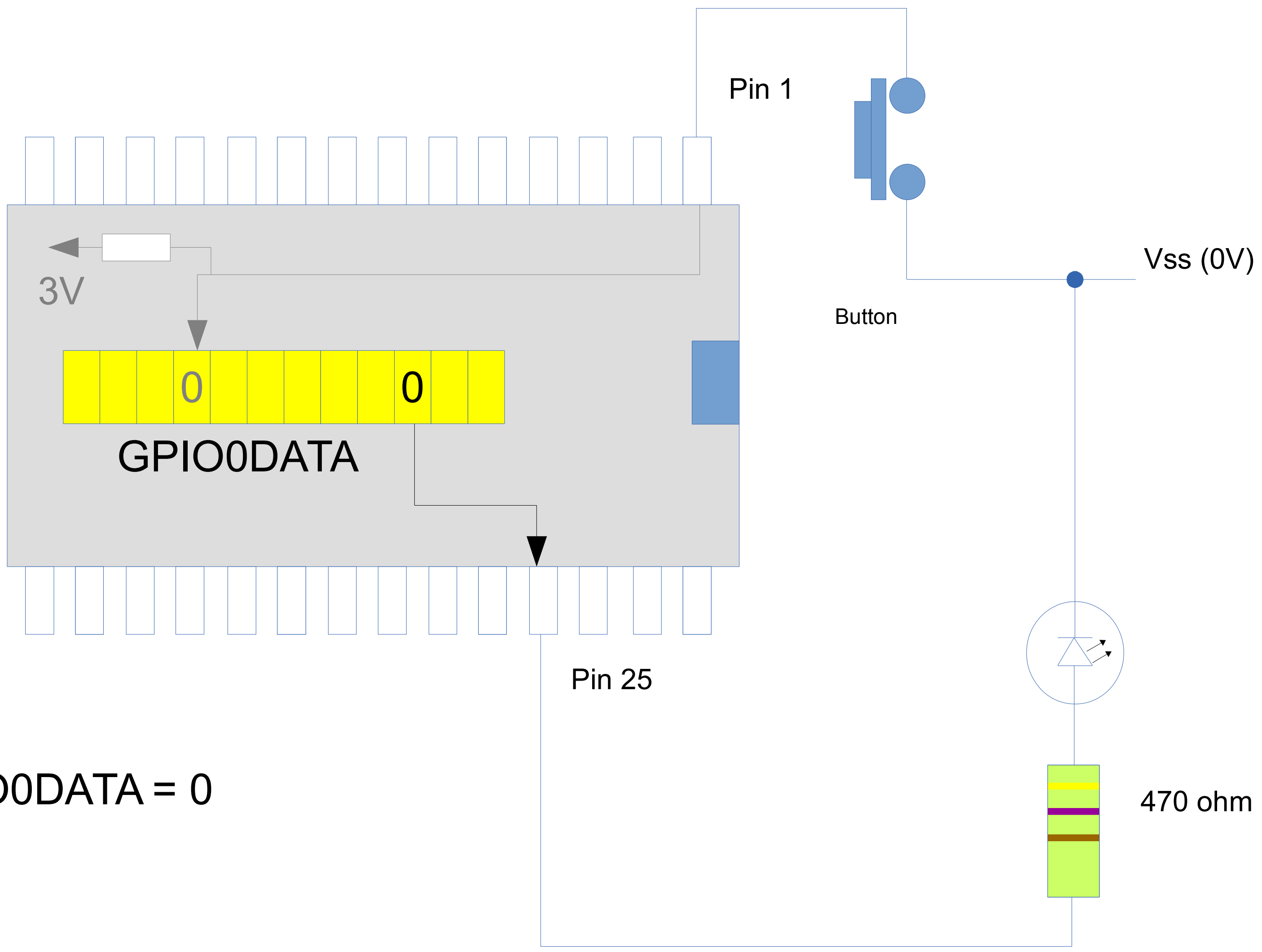
```
if (GPIO0DATA & BIT8)
{
        :
}
```

Parallel Input/Output



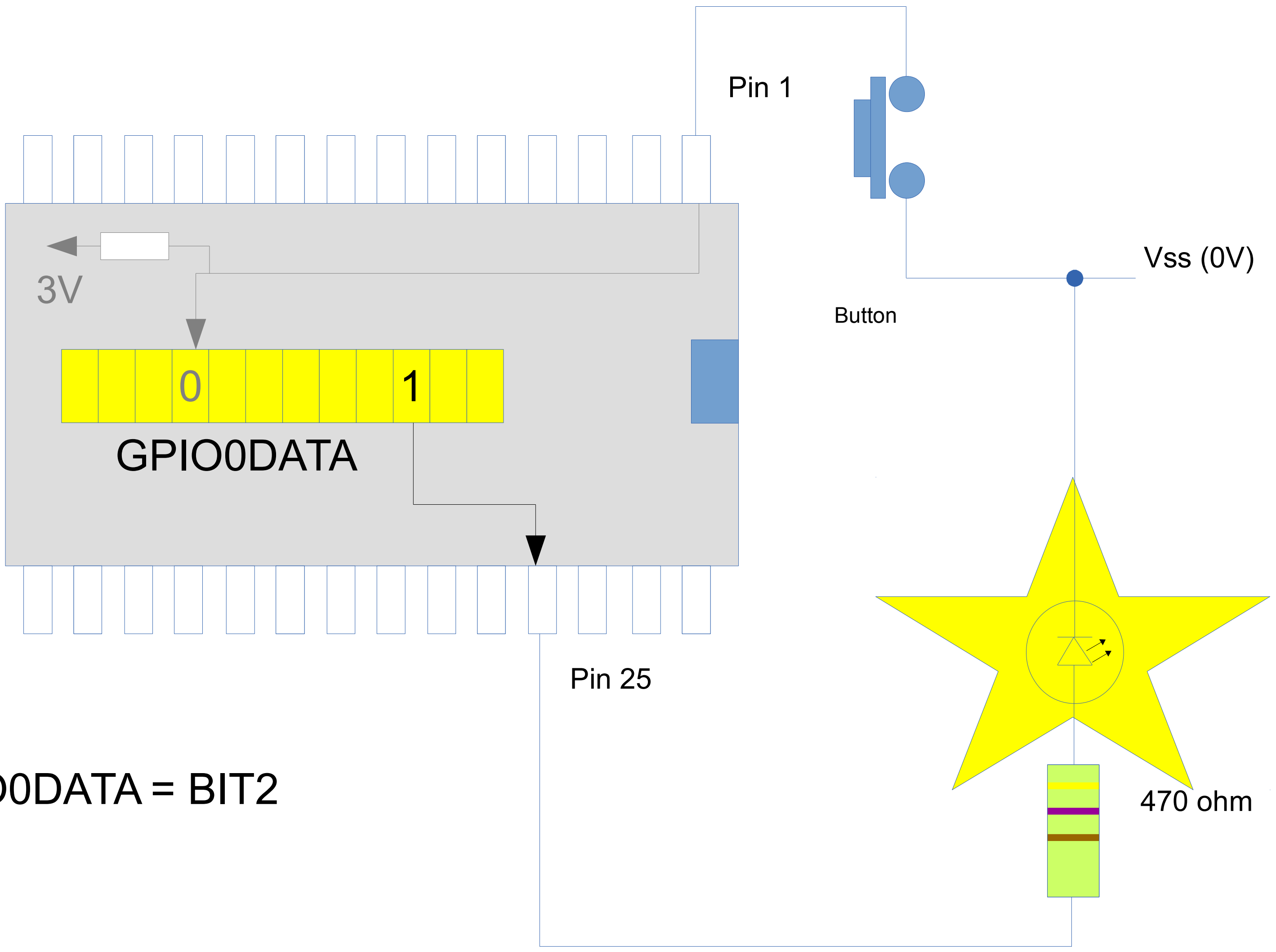
```
if (GPIO0DATA & BIT8)
{
        :
}
```

Parallel Input/Output



GPIO0DATA = 0

Parallel Input/Output



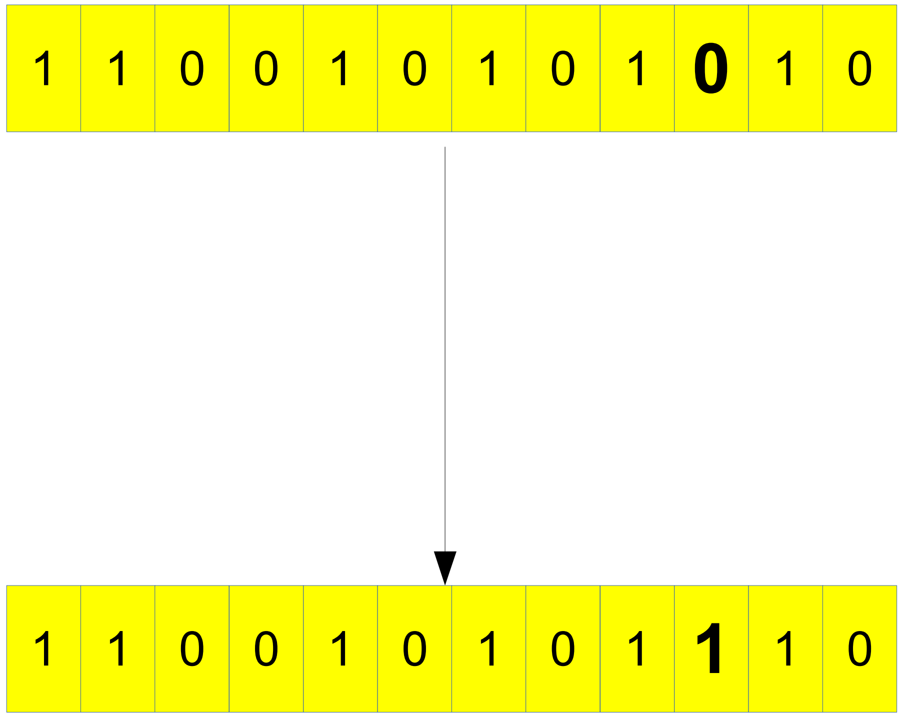
GPIO0DATA = BIT2

Parallel Input/Output

- LPC1114 Ports can have up to 12 bits
- C does not allow individual bit writes/reads
- We need to use **MASKS**

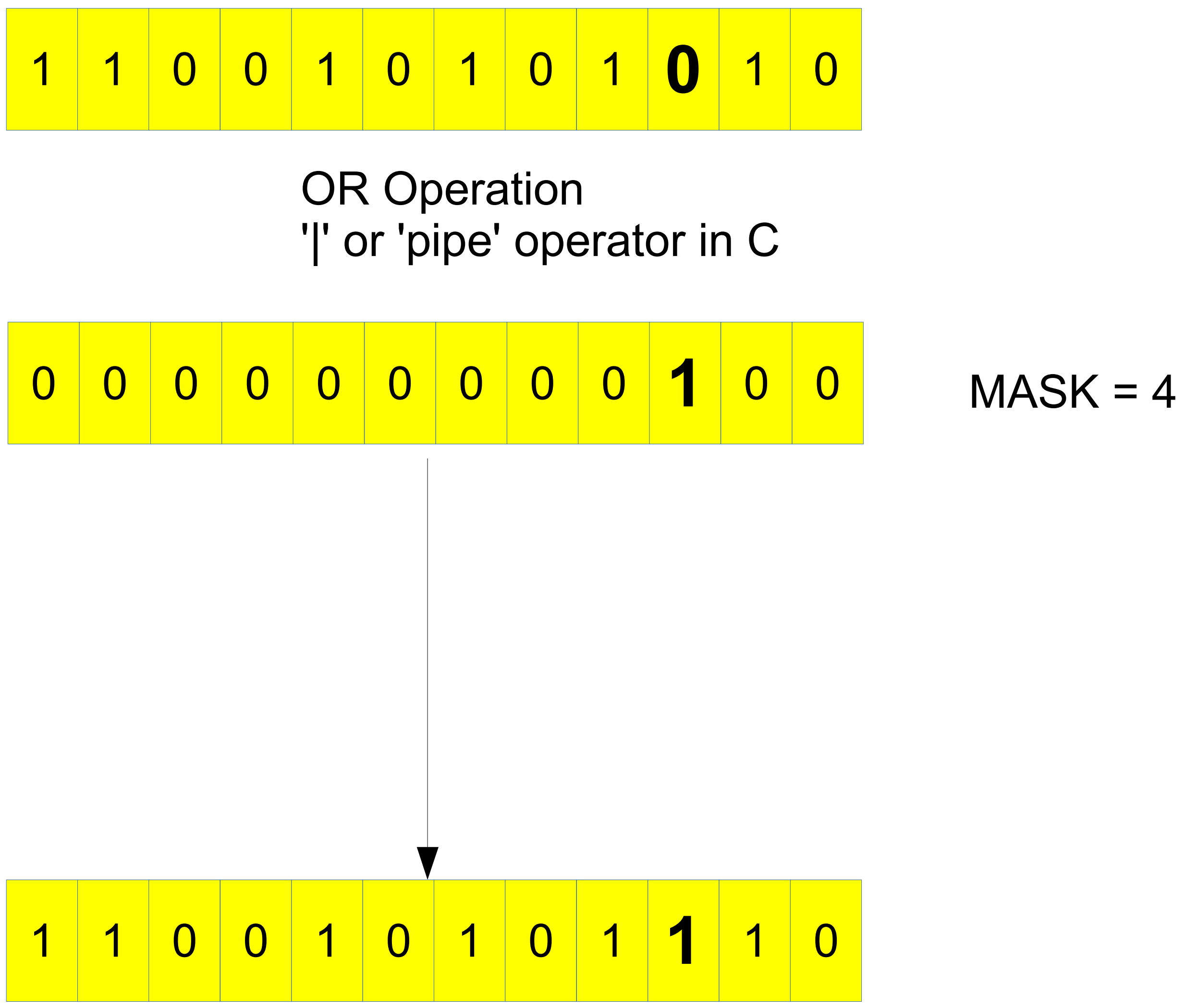
Parallel Input/Output

- Set BIT2 without affecting other bits
- Don't know state of other bits in advance



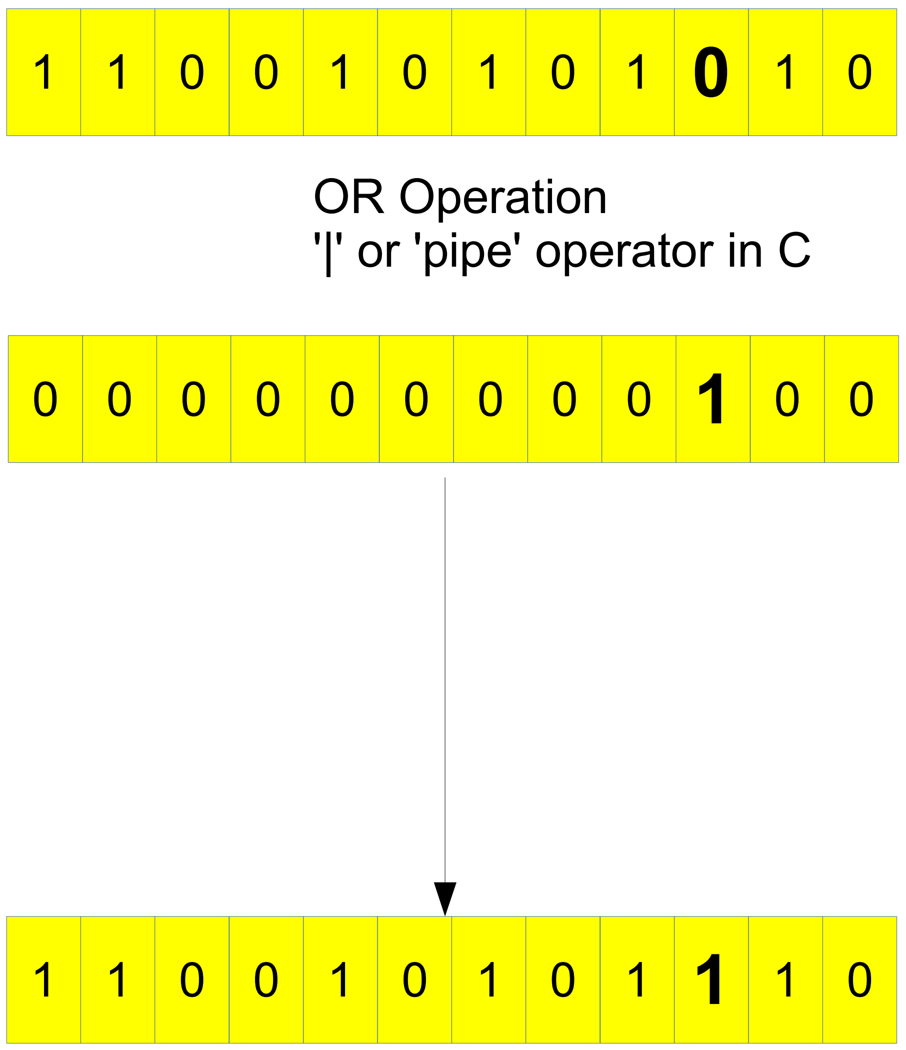
Parallel Input/Output

Set a bit



Parallel Input/Output

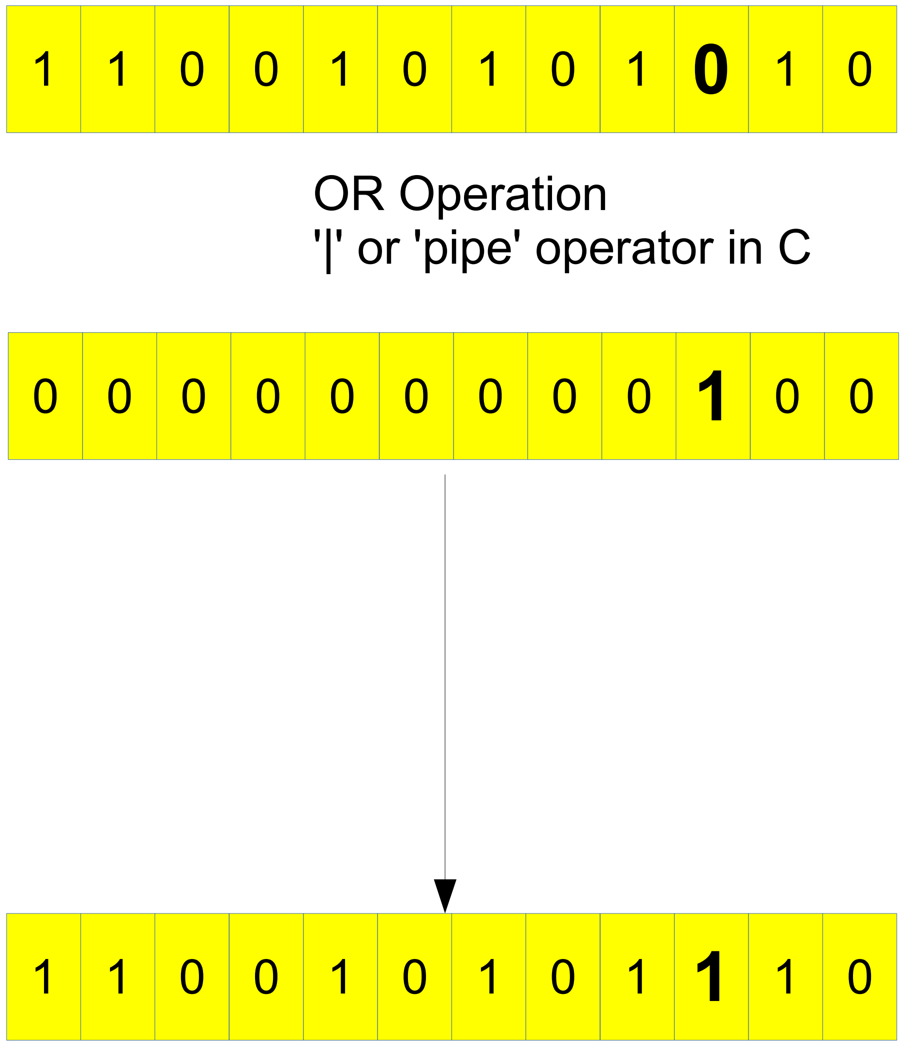
Set a bit



MASK = (1 << 2)

Parallel Input/Output

Set a bit



```
#define BIT2 (1 << 2)
```

OR Operation
'|' or 'pipe' operator in C

MASK = BIT2

Parallel Input/Output

Set a bit

GPIO0DATA

1	1	0	0	1	0	1	0	1	**0**	1	0

#define BIT2 (1 << 2)

OR Operation
'|' or 'pipe' operator in C

0	0	0	0	0	0	0	0	0	**1**	0	0

MASK = BIT2

GPIO0DATA = GPIO0DATA | BIT2

GPIO0DATA

1	1	0	0	1	0	1	0	1	**1**	1	0

Parallel Input/Output

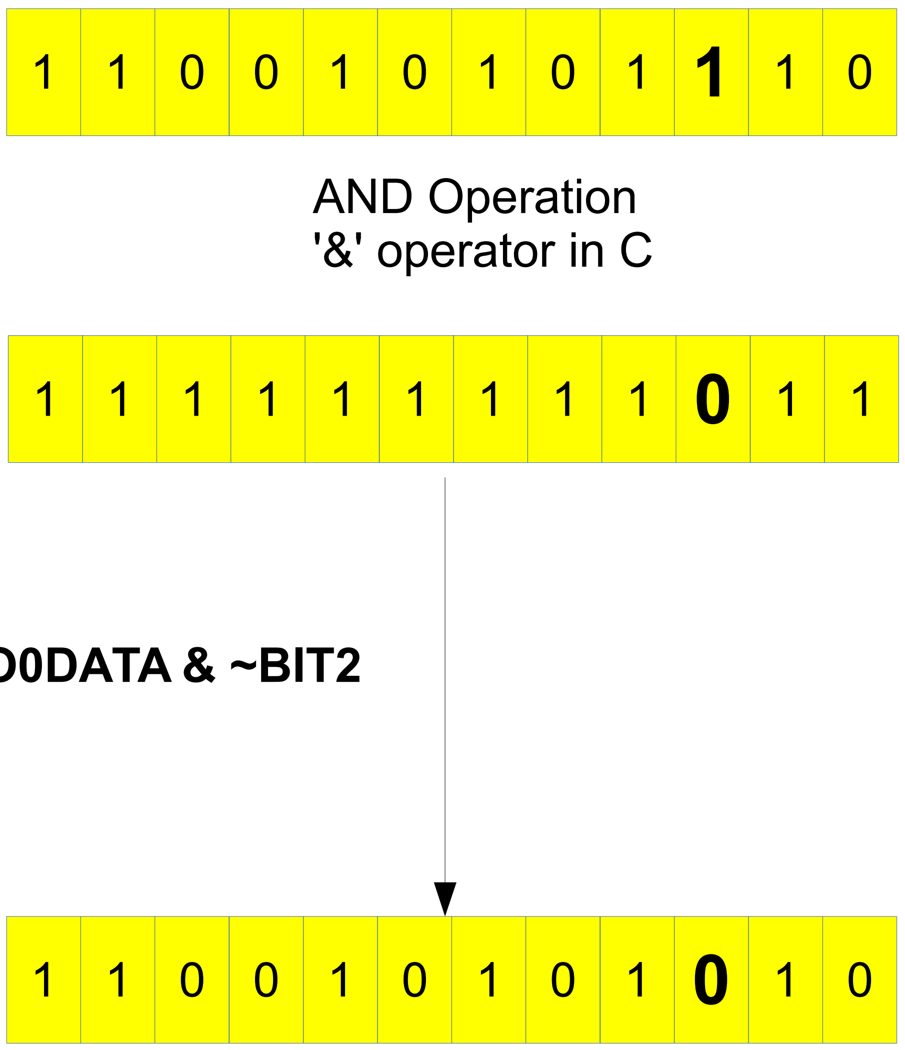
Clear a bit

GPIO0DATA

#define BIT2 (1 << 2)

AND Operation
'&' operator in C

MASK = ~BIT2



GPIO0DATA = GPIO0DATA & ~BIT2

GPIO0DATA

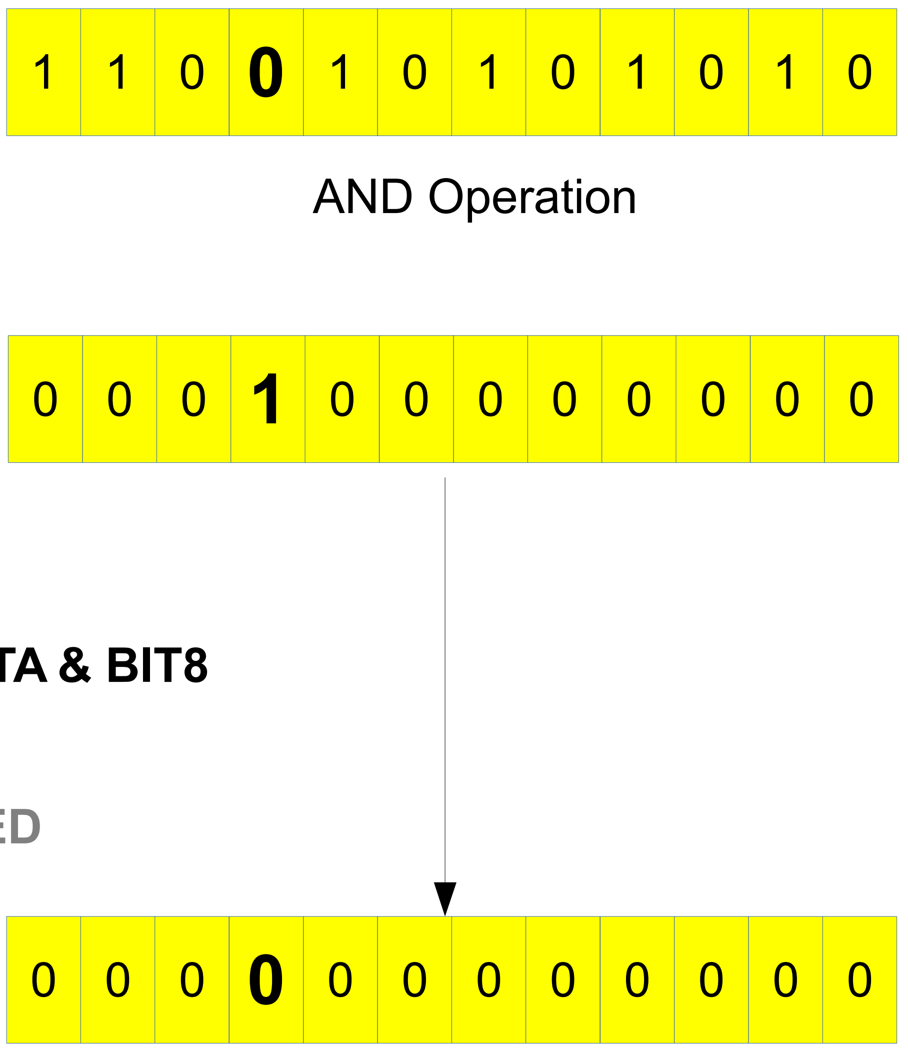
Parallel Input/Output

- Masks must also be used to test bits
- Useful for testing whether an input is high or low

Parallel Input/Output

Test a bit

GPIO0DATA



#define BIT2 (1 << 8)

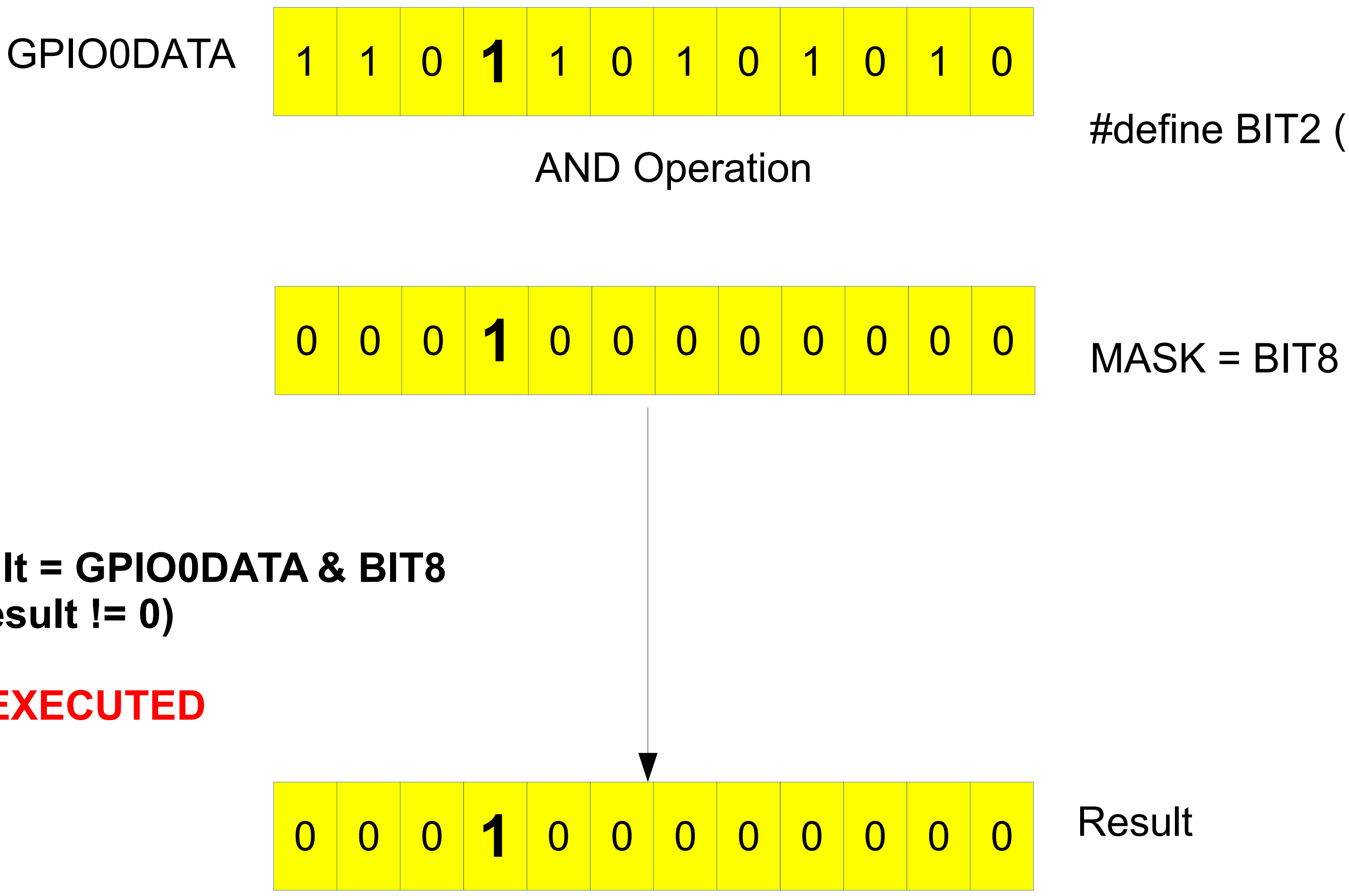
MASK = BIT8

Result

```
Result = GPIO0DATA & BIT8
if (Result != 0)
{
    NOT EXECUTED
}
```

Parallel Input/Output

Test a bit



#define BIT2 (1 << 8)

Result = GPIO0DATA & BIT8
if (Result != 0)
{
EXECUTED
}

Lab2: Blinky

```
/* Blinky with a button.
   An LED is attached to Pin 25 and a button is attached to
Pin 1
  When the button is pressed, the LED should blink

*/

#include "lpc111x.h"

void delay(unsigned len)
{
   while(len--);
}
```

Lab2: Blinky

```
void ConfigPins()
{
   SYSAHBCLKCTRL |= BIT6 + BIT16; // Turn on clock for GPIO and IOCON
   IOCON_PIO0_2 &= ~(BIT1+BIT0);  // ensure Pin 25 behaves as GPIO
   GPIO0DIR |= BIT2;   // Make Pin 25 an output
   GPIO0DIR &= ~BIT8;  // Make Pin 0 an input
   GPIO0DATA = 0;      // 0 output initially
}
```

Lab2: Blinky

```
int main()
{

    ConfigPins();

    while(1)
    {
        if (GPIO0DATA & BIT8)
        {
            GPIO0DATA ^= BIT2;
            delay(1000000);
        }

    }
}
```

Lab2: Blinky

Extract from lpc111x.h

```
:
// Macros to reduce typing later on
#define  REGISTER_32(ADDRESS) (*((volatile unsigned int *)(ADDRESS)))
:
:
// AHB Peripherals
#define GPIO0_BASE     0x50000000
#define GPIO1_BASE     0x50010000
#define GPIO2_BASE     0x50020000
#define GPIO3_BASE     0x50030000

// not dealing with mask registers here
#define GPIO0DATA      REGISTER_32(GPIO0_BASE + 0x3ffc)
#define GPIO0DIR       REGISTER_32(GPIO0_BASE + 0x8000)
```