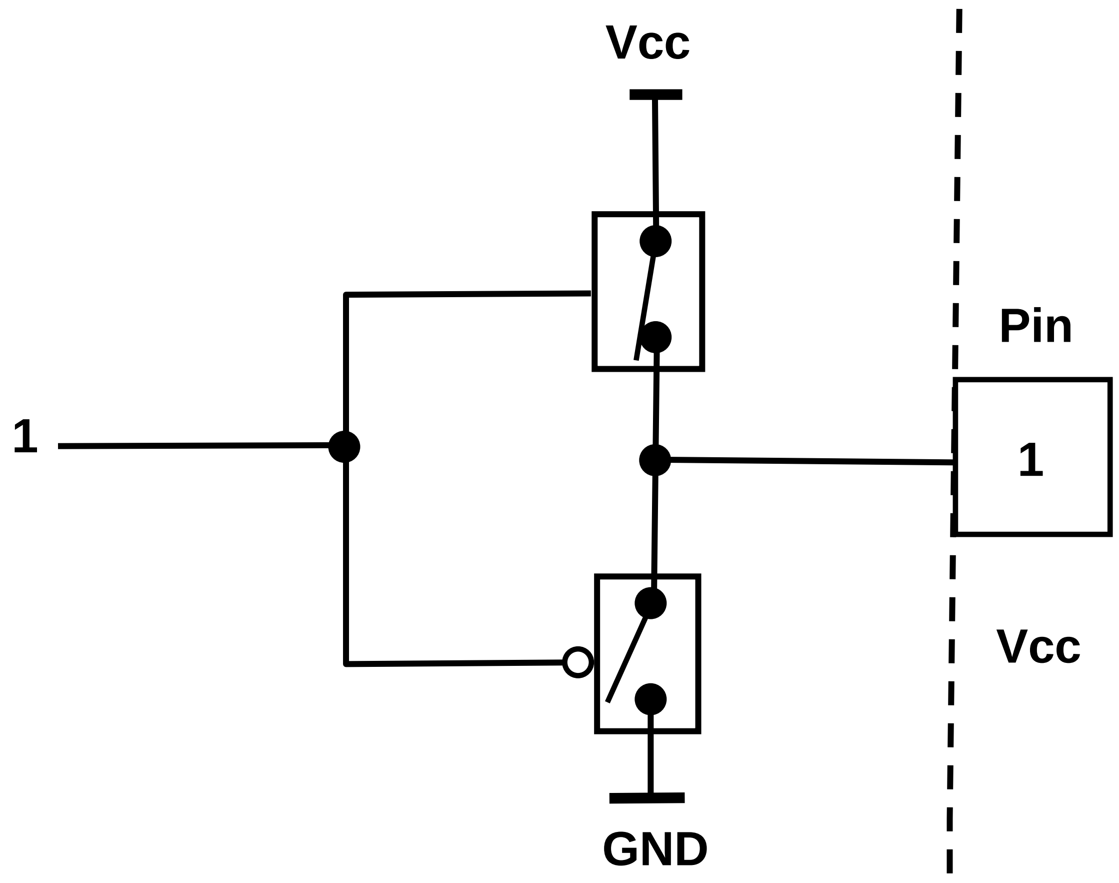
Microcontroladores

AVR Atmega - Puertos E/S

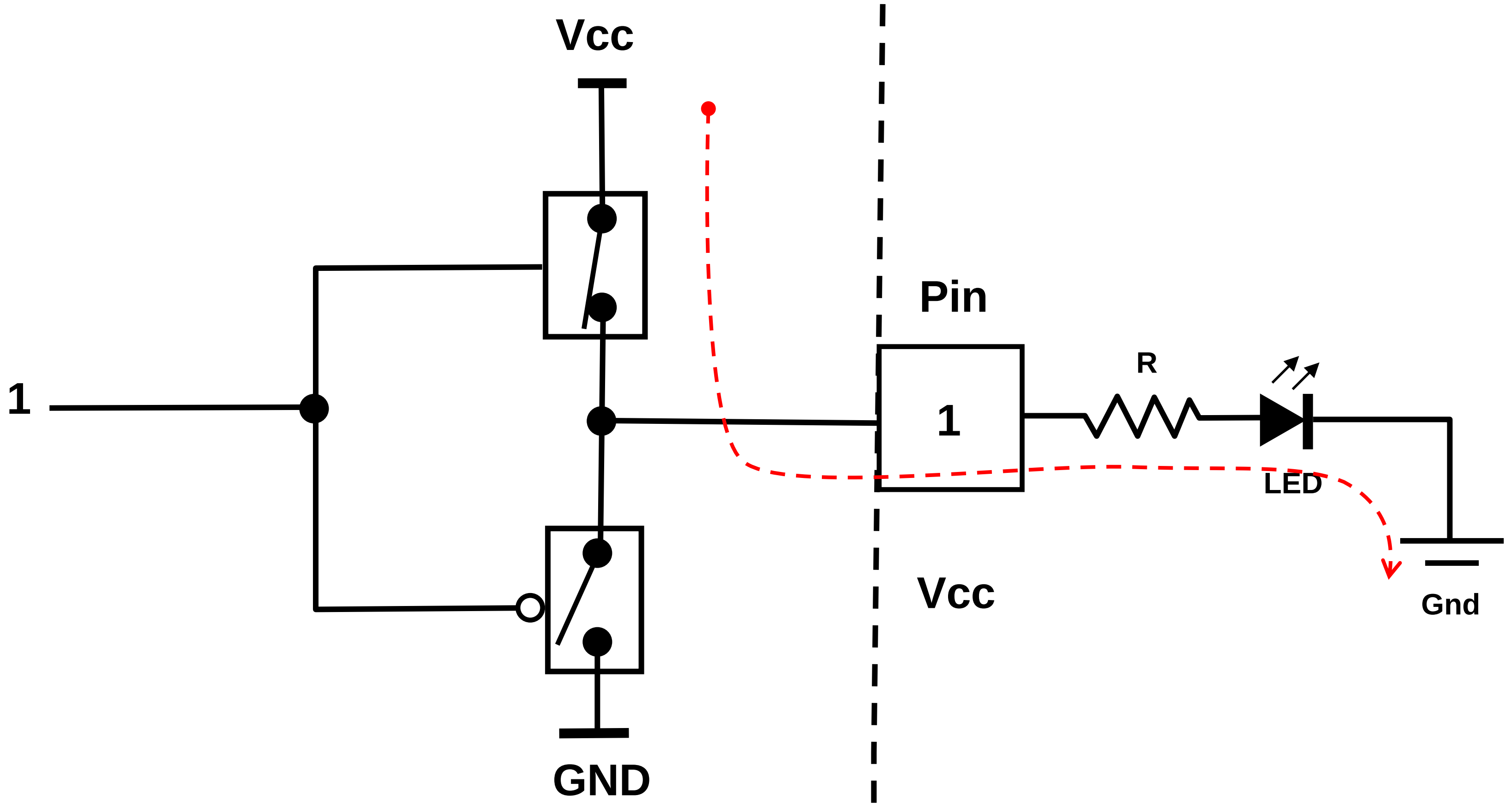
Características de los puertos E/S

- Manejadores Push-Pull
- Manejador de Alta corriente (hasta 20 mA)
- Controlador para resistencias Pull-Up (por pin)
- Controlador de dirección (por pin)
- Tres bits de control/estado por bit/pin
- Acceso tipo Read-Modify-Write

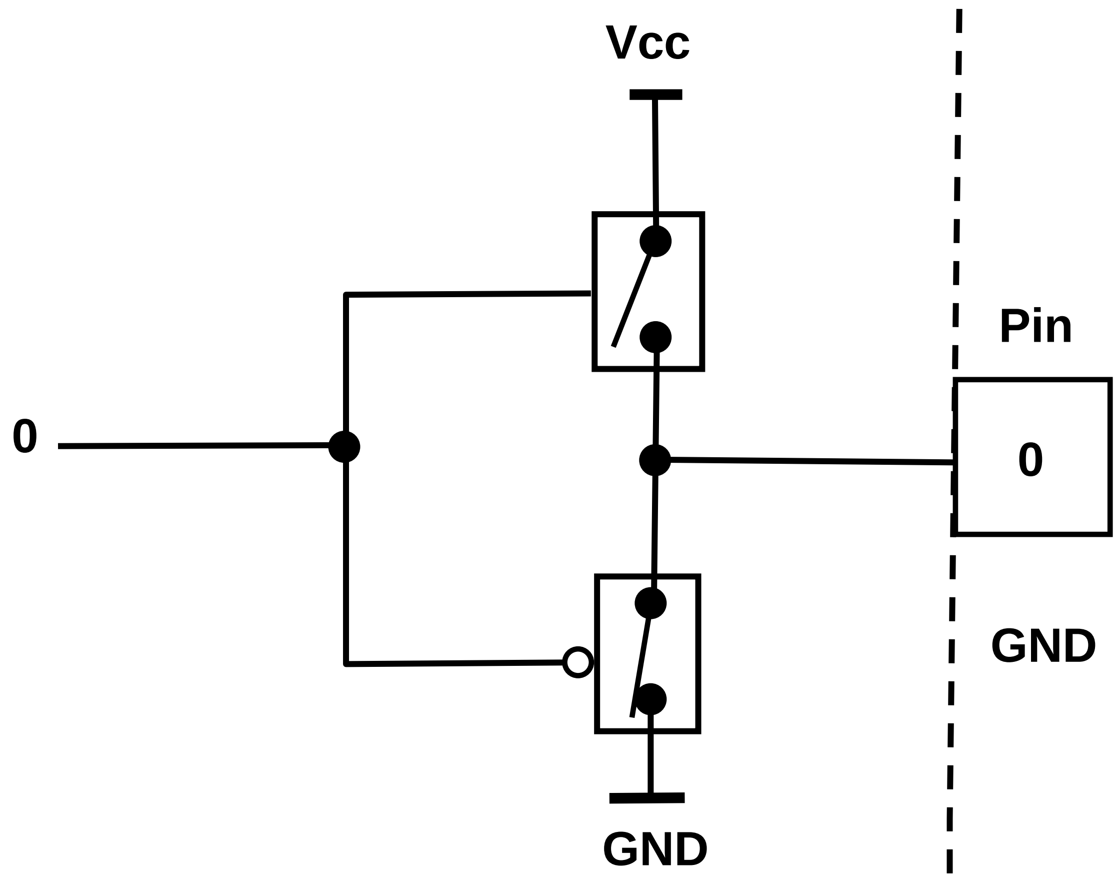
Manejador de Salida Push-Pull



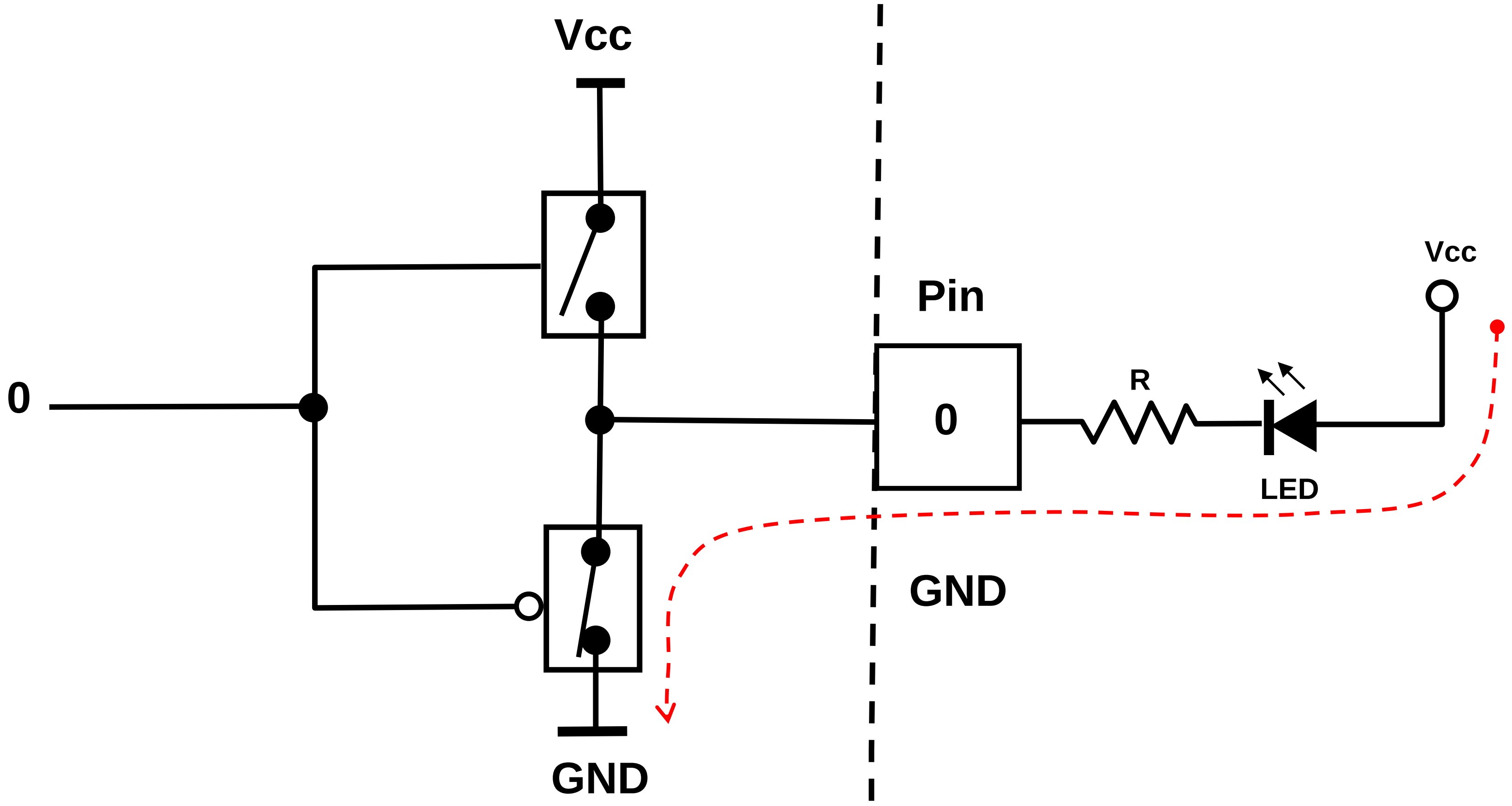
Manejador de Salida Push-Pull



Manejador de Salida Push-Pull



Manejador de Salida Push-Pull

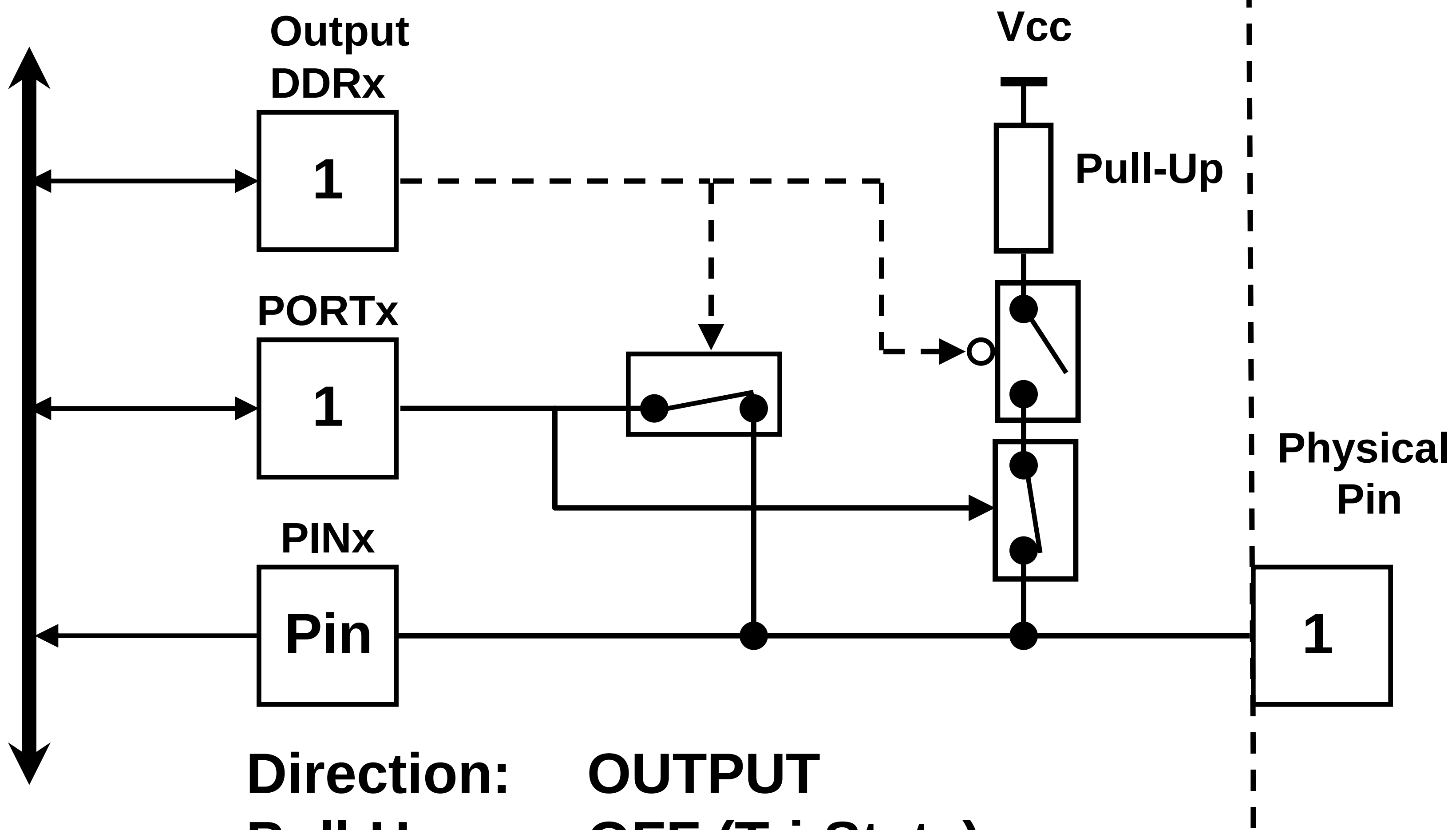


Bits de Control/Estado por Pin

DDRx	Data Direction Control Bit
PORTx	Output Data or Pull-Up Control Bit
PINx	Pin Level Bit

X = A, B, C, ...

Diagrama de bloques de un pin de E/S



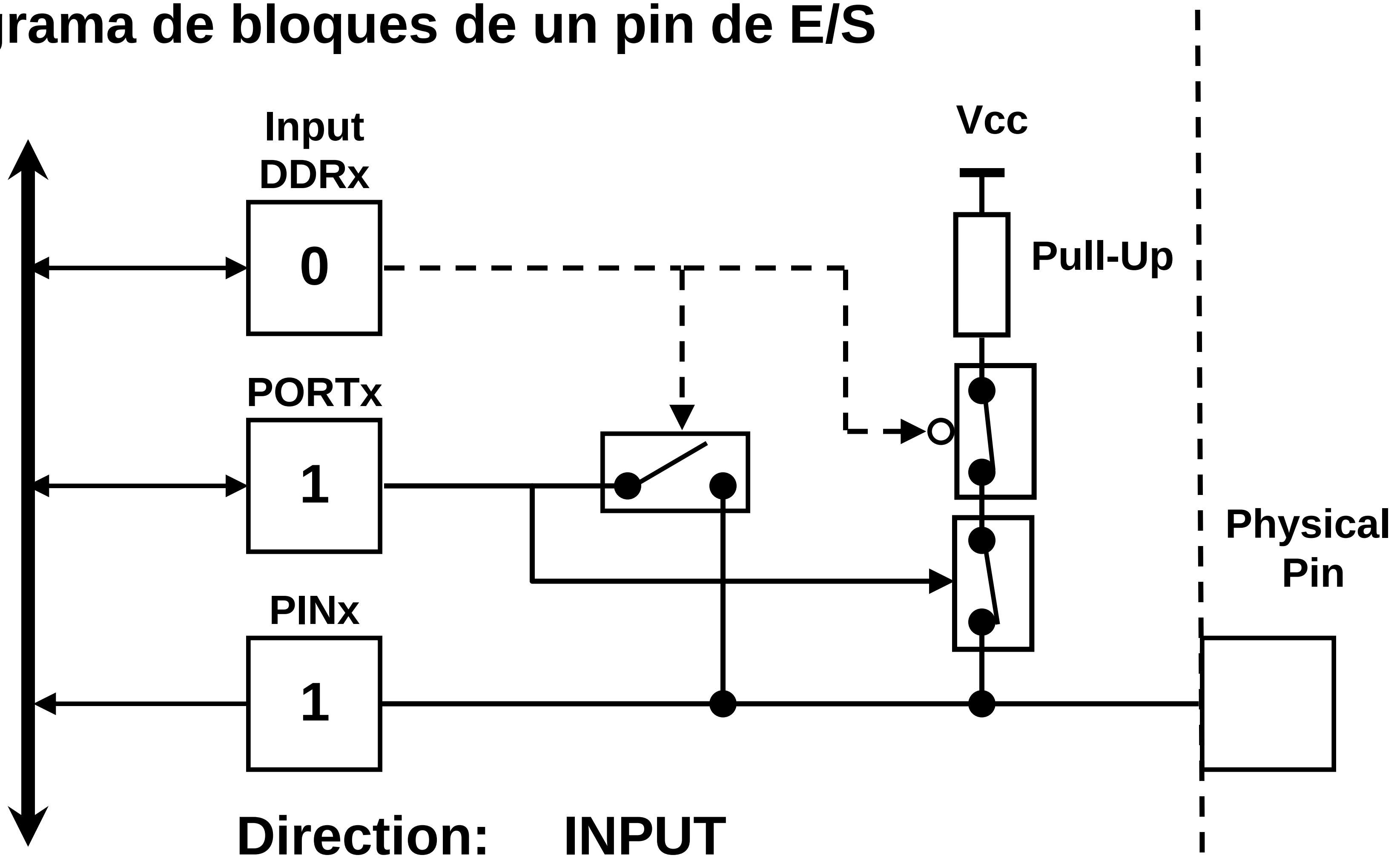
Direction: OUTPUT

Pull-Up: OFF (Tri-State)

Physical Pin x = PORTx

Pin x = Physical Pin x

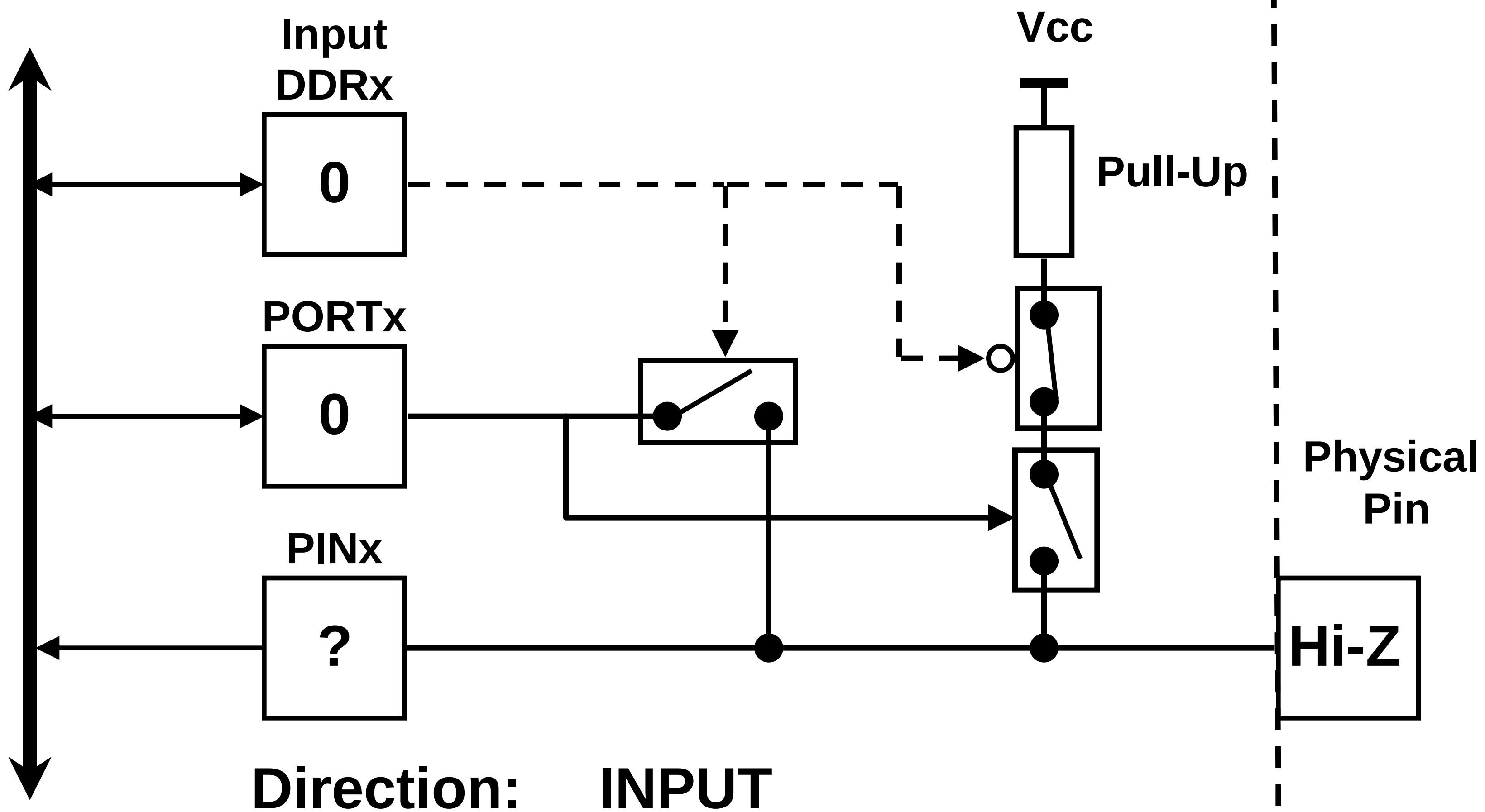
Diagrama de bloques de un pin de E/S



Direction: INPUT

Pull-Up: Activo (PORTx = 1)

Diagrama de bloques de un pin de E/S



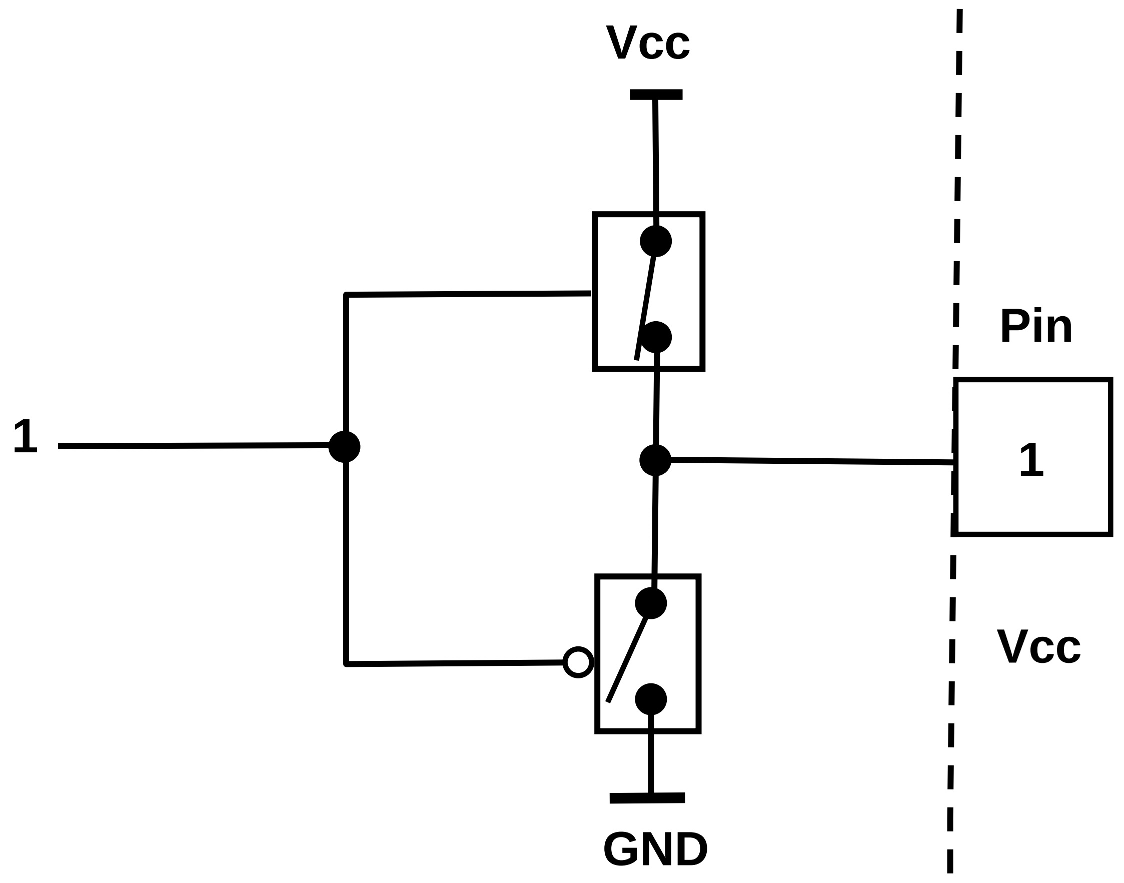
Direction: INPUT

Pull-Up: No activo (PORTx = 0)

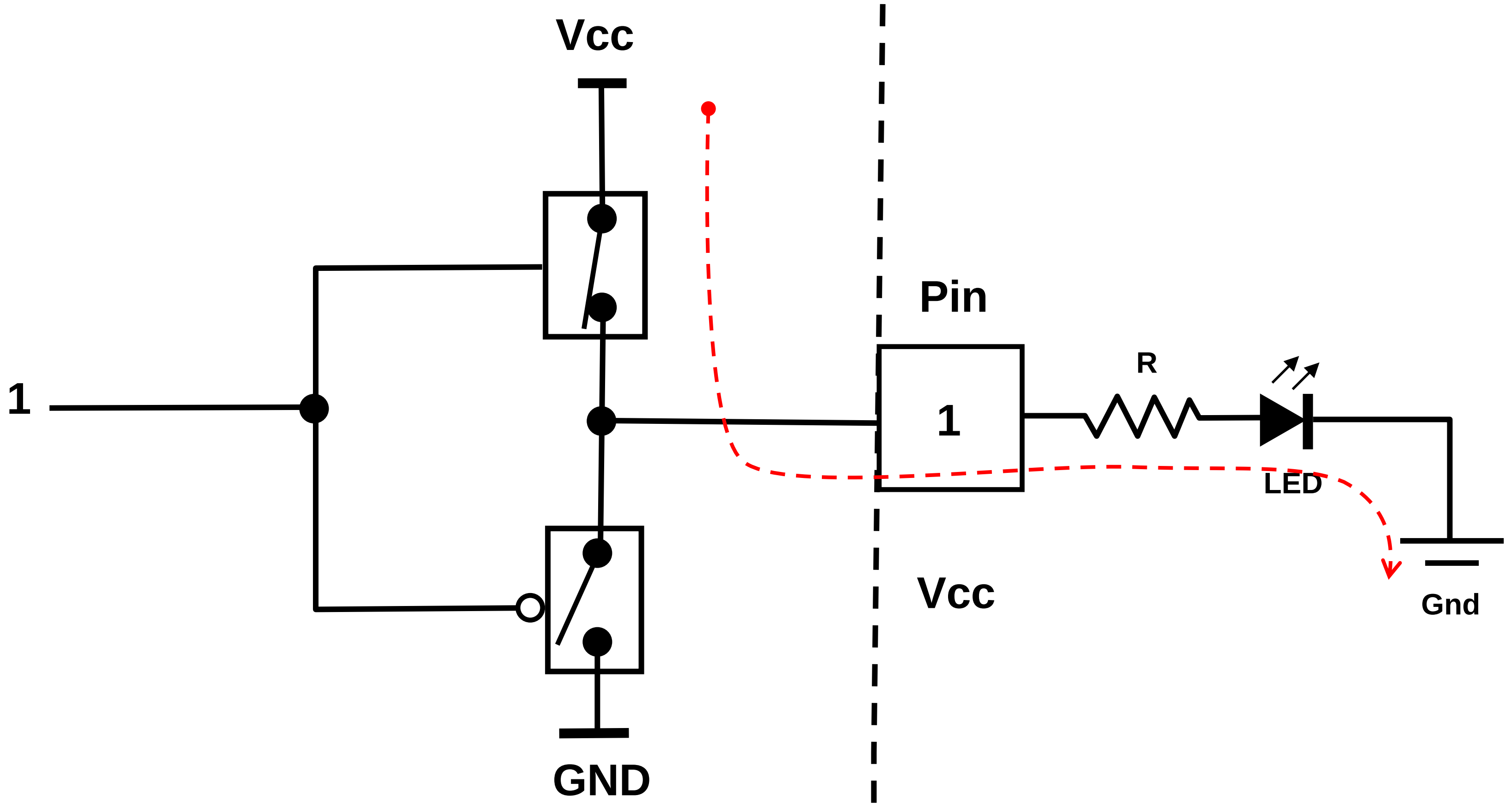
Tabla según bits de configuración

DDxn	PORTxn	PUD (in MCUCR)	I/O	Pull-up	Comment
0	0	X	Input	No	Tri-state (Hi-Z)
0	1	0	Input	Yes	Pxn will source current if ext. pulled low
0	1	1	Input	No	Tri-state (Hi-Z)
1	0	X	Output	No	Output Low (Sink)
1	1	X	Output	No	Output High (Source)

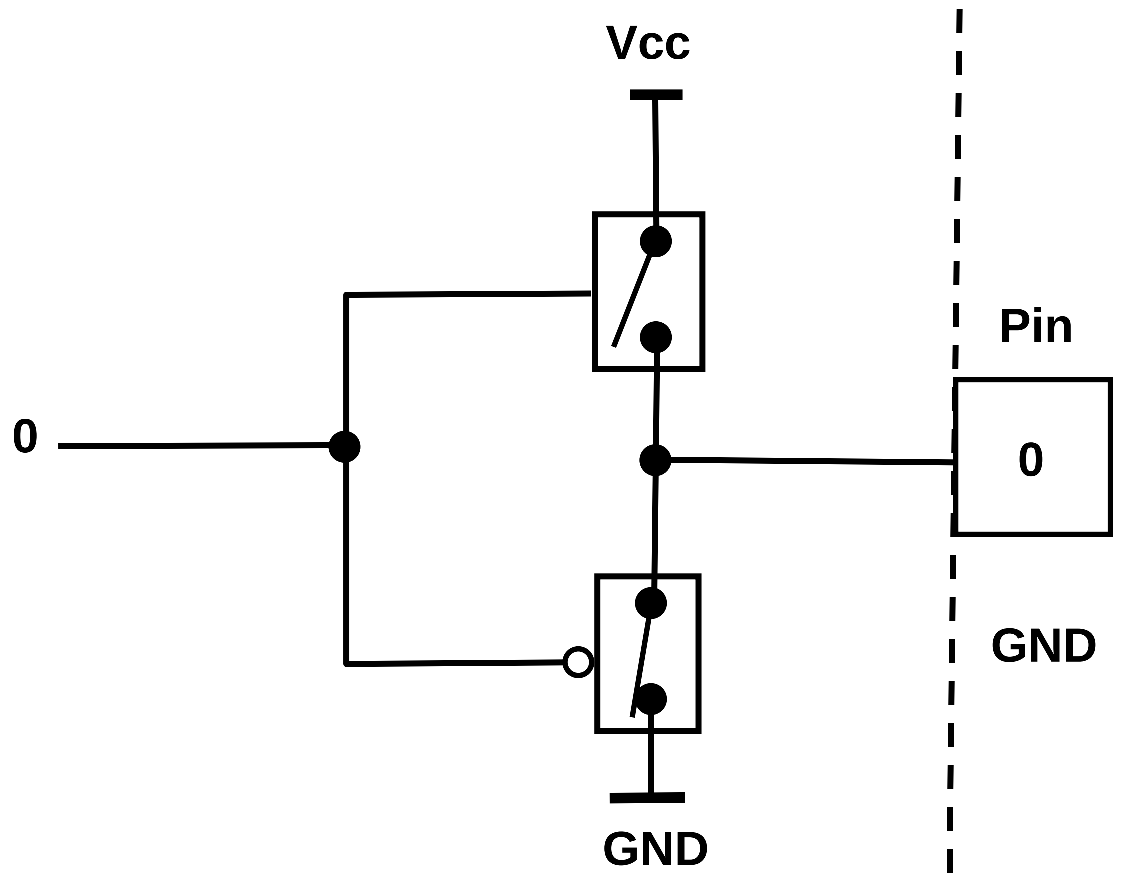
Manejador de Salida Push-Pull



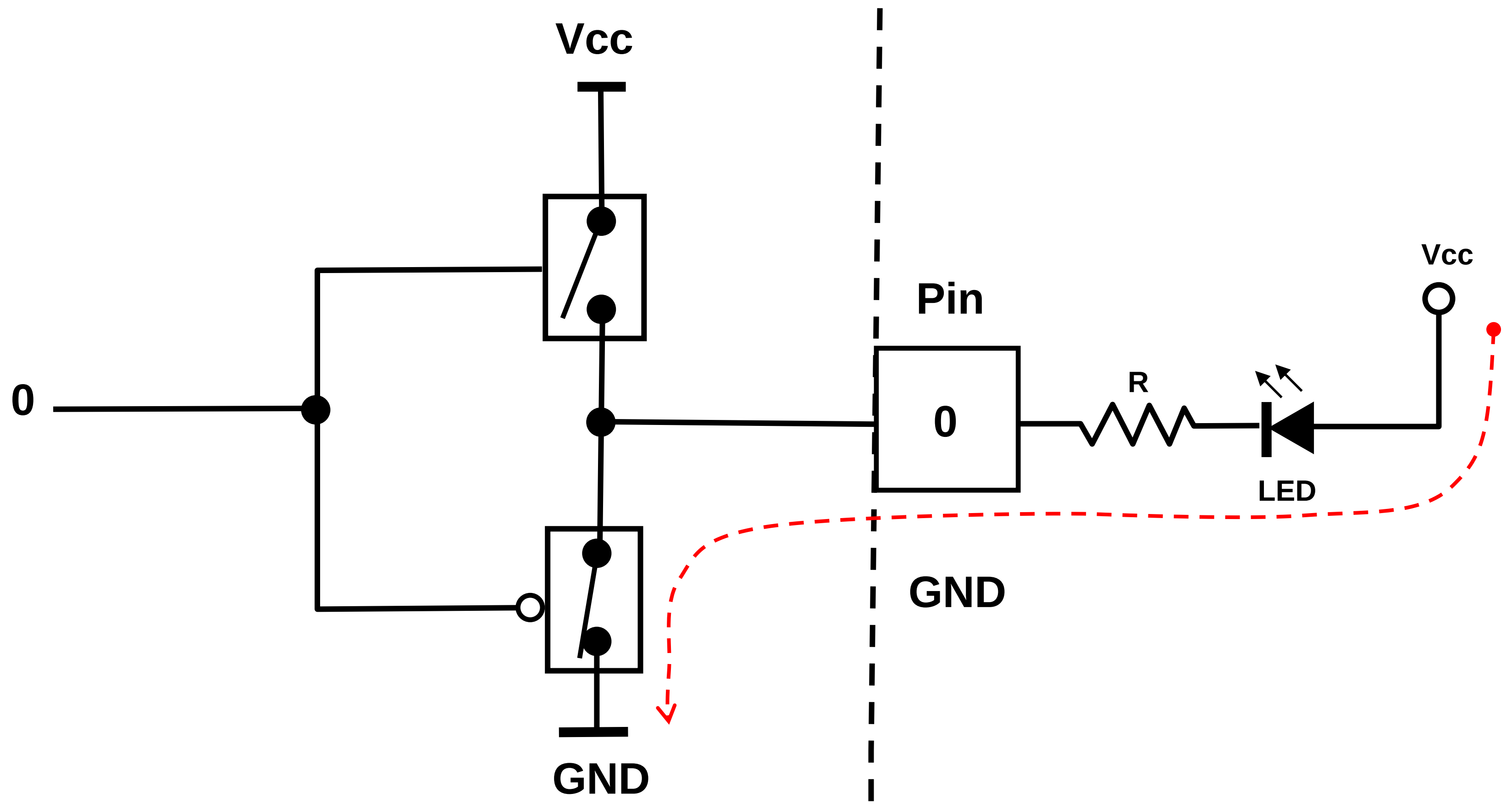
Manejador de Salida Push-Pull



Manejador de Salida Push-Pull



Manejador de Salida Push-Pull



Función SetBitPort(Puerto , nbit)

```
SetBitPort( &PortX, 5 );
```

```
+00000093:   E2E5       LDI    R30,0x25      Load immediate
+00000094:   E0F0       LDI    R31,0x00      Load immediate
+00000095:   8180       LDD    R24,Z+0       Load indirect with displacement
+00000096:   E065       LDI    R22,0x05      Load immediate
+00000097:   940E00A0   CALL   0x000000A0    Call subroutine
```

```
void SetBitPort ( unsigned char *Puerto, unsigned char nbit){
   Puerto = Puerto | ( 1 << nbit );
}
```

```
23:        void SetBitPort ( unsigned char *Puerto, unsigned char nbit){
+0000009E:   93DF        PUSH    R29              Push register on stack
+0000009F:   93CF        PUSH    R28              Push register on stack
+000000A0:   D000        RCALL   PC+0x0001        Relative call subroutine
+000000A1:   920F        PUSH    R0               Push register on stack
+000000A2:   B7CD        IN      R28,0x3D         In from I/O location
+000000A3:   B7DE        IN      R29,0x3E         In from I/O location
+000000A4:   839A        STD     Y+2,R25          Store indirect with displacement
+000000A5:   8389        STD     Y+1,R24          Store indirect with displacement
+000000A6:   836B        STD     Y+3,R22          Store indirect with displacement
24:            *Puerto = *Puerto | ( 1 << nbit );
+000000A7:   81E9        LDD     R30,Y+1          Load indirect with displacement
+000000A8:   81FA        LDD     R31,Y+2          Load indirect with displacement
+000000A9:   8180        LDD     R24,Z+0          Load indirect with displacement
+000000AA:   2F48        MOV     R20,R24          Copy register
+000000AB:   818B        LDD     R24,Y+3          Load indirect with displacement
+000000AC:   2F28        MOV     R18,R24          Copy register
+000000AD:   E030        LDI     R19,0x00         Load immediate
+000000AE:   E081        LDI     R24,0x01         Load immediate
+000000AF:   E090        LDI     R25,0x00         Load immediate
+000000B0:   2E02        MOV     R0,R18           Copy register
+000000B1:   C002        RJMP    PC+0x0003        Relative jump
+000000B2:   0F88        LSL     R24              Logical Shift Left
+000000B3:   1F99        ROL     R25              Rotate Left Through Carry
+000000B4:   940A        DEC     R0               Decrement
+000000B5:   F7E2        BRPL    PC-0x03          Branch if plus
+000000B6:   2B84        OR      R24,R20          Logical OR
+000000B7:   81E9        LDD     R30,Y+1          Load indirect with displacement
+000000B8:   81FA        LDD     R31,Y+2          Load indirect with displacement
+000000B9:   8380        STD     Z+0,R24          Store indirect with displacement
25:        }
+000000BA:   900F        POP     R0               Pop register from stack
+000000BB:   900F        POP     R0               Pop register from stack
+000000BC:   900F        POP     R0               Pop register from stack
+000000BD:   91CF        POP     R28              Pop register from stack
+000000BE:   91DF        POP     R29              Pop register from stack
+000000BF:   9508        RET                      Subroutine return
```

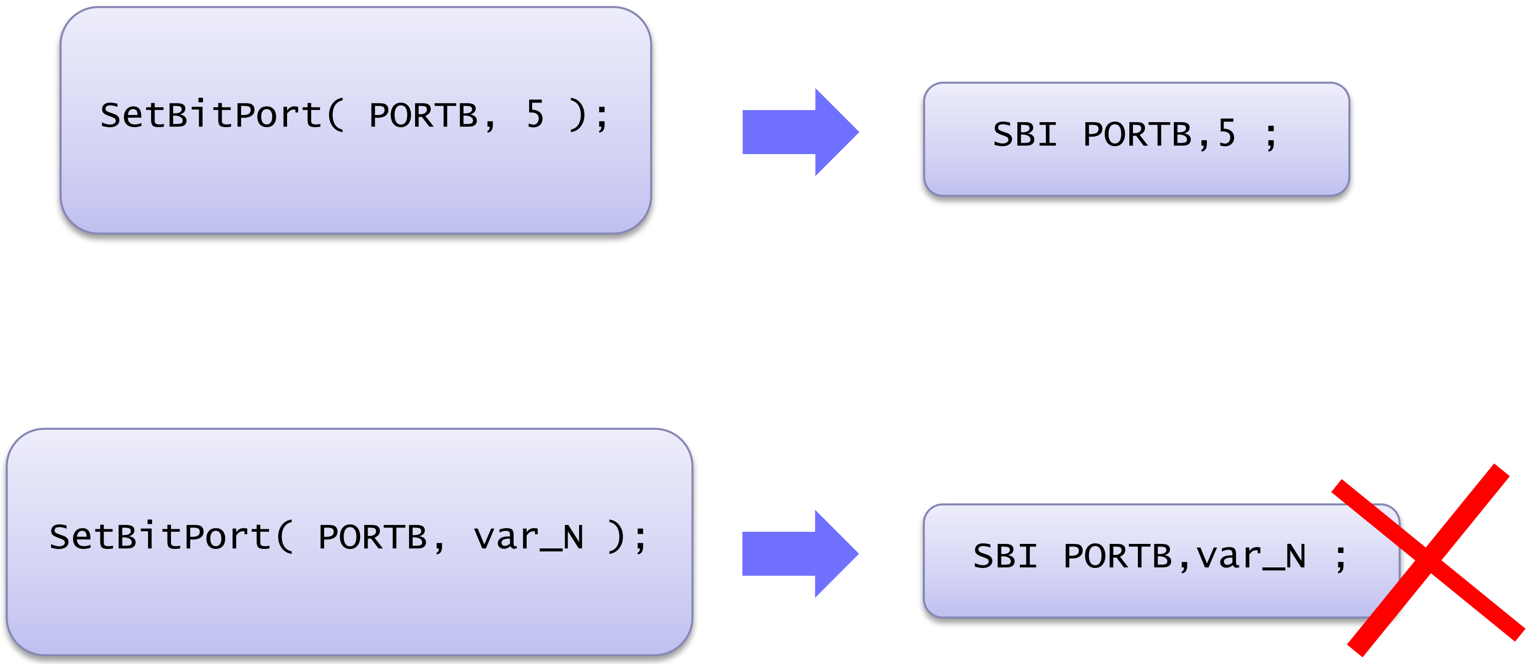
MACRO - Assembler inline

```
SetBitPort( PORTB, 5 );
```

$\Rightarrow$

```
SBI PORTB,5 ;
```

MACRO - Assembler inline



```
SetBitPort( PORTB, 5 );
```

```
SBI PORTB,5 ;
```

```
SetBitPort( PORTB, var_N );
```

```
SBI PORTB,var_N ;
```