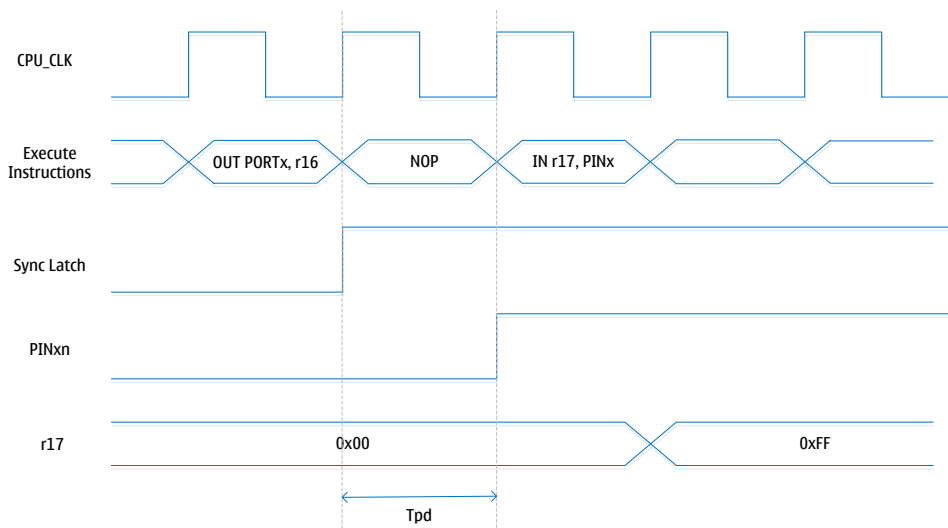


存，并且在下一个时钟的上升沿被寄存器到 **PINxn** 寄存器。上图中的 **Tpd,max** 以及 **Tpd,min** 为端口数据的最大和最小延迟，分为为 **1.5** 周期和 **0.5** 周期。

如果要读取到软件设置的端口值，需要在 **I/O** 的写和读字节支持插入一个空操作指令 (**NOP**)。时序如下图所示：



下面的代码说明如何设置端口 **B** 的引脚 **0/1** 为高，**2/3** 为低，定义引脚 **4~7** 为输入并且使能了引脚 **6、7** 的上拉电阻。然后引脚的值回读到通用工作寄存器中，按照之前的描述，在引脚的输出和输入直接插入了一个 **NOP** 指令。

#### 汇编代码

```
; Define Pull-ups and set outputs high
; Define directions for port pins
LDI r16, (1<<PB7)|(1<<PB6)|(1<<PB1)|1<<PB0)
LDI r17, (1<<DDB3)|(1<<DDB2)|(1<<DDB1)|(1<<DDB0)
OUT PORTB, r16
OUT DDRB, r17
; Insert nop for synchronization
NOP
; Read port pins
IN r16, PINB
```

#### C 语言代码

```
unsigned char I;
/* Define pull-ups and set outputs high */
/* Define directions for port pins */
PORTB = (1<<PB7)|(1<<PB6)|(1<<PB1)|(1<<PB0);
DDRB = (1<<DDB3)|(1<<DDB2)|(1<<DDB1)|(1<<DDB0);
/* Insert nop for synchronization */
__no_operation();
/* Read port pins */
I = PINB;
```