



Programcounter (CNT)

Labor Digital Design

Contents

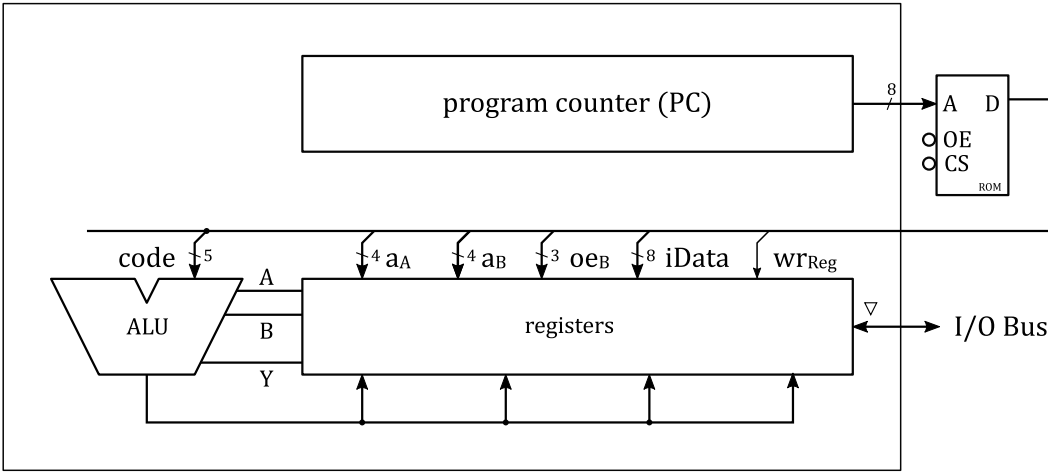
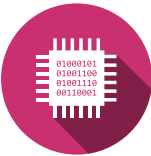


Figure 1:

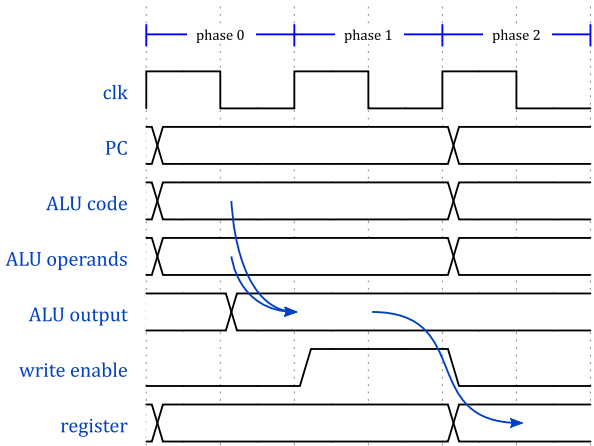


Figure 2:

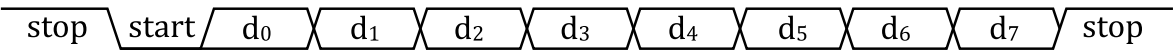


Figure 3:

```
LOAD      s3, FF          ; load stop bit
OUTPUT    s3              ; output stop bit
LOAD      s3, s3          ; no operation
LOAD      s3, s3          ; no operation
LOAD      s3, s3          ; no operation
LOAD      s3, s3          ; no operation
LOAD      s0, 00          ; load start bit
OUTPUT    s0              ; output start bit
INPUT     s1              ; load word to send
OUTPUT    s1              ; output word, LSB is considered
SR0       s1              ; shift word, bit 1 -> LSB
OUTPUT    s1              ; output bit 1
SR0       s1              ; bit 2 -> LSB
OUTPUT    s1              ; output bit 2
SR0       s1              ; bit 3 -> LSB
OUTPUT    s1              ; output bit 3
SR0       s1              ; bit 4 -> LSB
OUTPUT    s1              ; output bit 4
SR0       s1              ; bit 5 -> LSB
```



```
OUTPUT    s1                ; output bit 5
SR0       s1                ; bit 6 -> LSB
OUTPUT    s1                ; output bit 6
SR0       s1                ; bit 7 -> LSB
OUTPUT    s1                ; output bit 7
LOAD      s3, s3            ; no operation
OUTPUT    s3                ; output stop bit
```

```
LOAD      s3, FF           ; load stop bit
OUTPUT    s3                ; output stop bit
LOAD      s2, 04           ; initialize loop counter 3
SUB       s2, 01           ; decrement loop counter 4
JUMP NZ   03              ; loop back if not end of count 5
LOAD      s0, 00           ; load start bit 6
OUTPUT    s0                ; output start bit 7
LOAD      s2, 08           ; initialize loop counter 8
INPUT     s1                ; load word to send 9
LOAD      s3, s3            ; no operation
OUTPUT    s1                ; output word, LSB is considered
SR0       s1                ; next bit -> LSB
SUB       s2, 01           ; decrement loop counter
JUMP NZ   0A              ; loop back if not end of count
OUTPUT    s3                ; output stop bit
```