

17 luglio 2024

A → ROM d. N lee da 16 bit

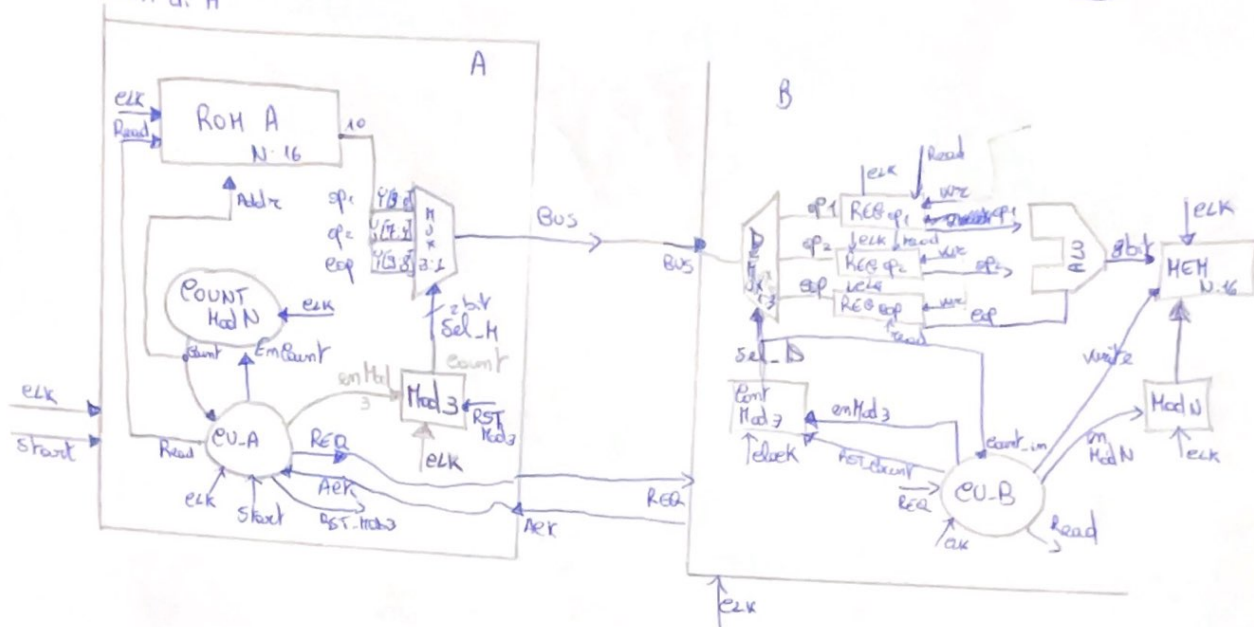
B → ROM d. N lee da 8 bit

ALU

100 S, 01 shift, 10 mult. pl, 11 divia

Il B affettua grazie: one e marmozza m H

OK



Hand-drawn state transition diagram for a bus protocol. The diagram shows the following states and transitions:

- Idle**: Initial state. Transitions: $start = 'x'$ to **Wake on Bus**; $start = '0'$ and $sd_H = 00$ to **Idle**.
- Wake on Bus**: Transitions: $Req = 'x'$ to **SEND REQ**.
- SEND REQ**: Labeled "APK=0" with a blue arrow. Transitions: $REQ = 'x'$ to **Wait-4-APK-1**.
- Wait-4-APK-1**: Labeled "APK='0'". Transitions: $APK = '0'$ to **Address REQ**.
- Address REQ**: Labeled "REQ='0'". Transitions: $APK = 'x'$ to **Wait-4-APK-0**.
- Wait-4-APK-0**: Labeled "APK='0'". Transitions: $APK = '0'$ to **CHECK_COUNT Mod 3**.
- CHECK_COUNT Mod 3**: Labeled "in Count Mod 3 = 'x'". Transitions: $count \neq '0'$ to **SEND REQ**; $count = '0'$ to **CHECK_COUNT Mod N**.
- CHECK_COUNT Mod N**: Labeled "RST-Mod 3 = 'x' in Count = 'x'". Transitions: $count \neq '0'$ to **SEND REQ**; $count = '0'$ to **Wait-4-APK-0**.

Handwritten notes on the right side of the diagram:

- Resta qui per mgn smitichera la comarca zione
- Parcha? Hp check d'voci

Resto qui
per non
smuovere
la camera
? Hp clock
diversi

```

graph LR
    S((Wait & Request)) -- "REQ = 1" --> T((Send ACK))
    T -- "ACK = 1" --> A((Accept ACK))
    A -- "count Mod 3 = 1" --> C((Check Count Mod 3))
    C -- "count Mod 3 = 1" --> R((Read Reg))
    R -- "Write in Mem" --> W((Write in Mem))
    W -- "Increment" --> I((Increment))
    I -- "count Mod 3 = 1" --> S
  
```

The diagram illustrates a mutual exclusion protocol with the following states and transitions:

- Wait & Request**: Initial state. Transition to **Send ACK** when $REQ = 1$.
- Send ACK**: State where the process sends an acknowledgment. Transition to **Accept ACK** when $ACK = 1$.
- Accept ACK**: State where the process accepts the acknowledgment. Transition to **Check Count Mod 3** when $count \text{ Mod } 3 = 1$.
- Check Count Mod 3**: State where the process checks the count modulo 3. Transition to **Read Reg** when $count \text{ Mod } 3 = 1$.
- Read Reg**: State where the process reads the register. Transition to **Write in Mem** when $Write = 1$.
- Write in Mem**: State where the process writes to memory. Transition to **Increment** when $Increment = 1$.
- Increment**: State where the process increments the counter. Transition back to **Wait & Request** when $count \text{ Mod } 3 = 1$.

I segnali relativi ai contatori si chiamano
il modo di clock successivo