Lecture Aids,  Project 5: SMICRO Instruction Execution Timing

CMPEN  471

Kyusun  Choi
Project 5: SMICRO Instruction Execution Timing
Project 5: SMICRO Instruction Execution Timing

10/8/2015
10/8/2015

Project 5: SMICRO Instruction Execution
Timing
CPC clock
RPC resetb

Simple Microcontroller (SMC)

CPU
Control Unit (CU)

Instruction Decoder (ID)

SM
RPC
dcpc
CPC
SDM
WDR
DOR

IRO
RW

ASB
doeb
web
csb

ROM
data

addr

ROM

idat

iadr

DMUX

SDM

mxout

WDR

DR

drout

nonINV BUF

DOE

Data Path (DP)

addr

RAM
data

mdat

Project 5: MICRO Instruction Execution
Timing

10/8/2015
Project 5: SMICRO Instruction Execution Timing

10/8/2015
Project 5: SMicro Instruction Execution Timing

10/8/2015
Project 5: SMICRO Instruction Execution Timing
CPC (Clock for Program Counter) = clock
RPC (Reset for Program Counter) = resetb
T (instruction clock) <= clock/2
iadr (instruction/program address)
idat (instruction op-code and operand)
CPC clock
RPC resetb
T

iadr
00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21

idat
00 00 64 09 60 05 64 08 60 07 96 05 96 07 X X X X X X X X X X X X X 96 05

Project 5: SMICRO Instruction Execution
Timing
NOP
LD  #9
ST  5
LD  #8
ST  7
LD  5
LD  7

10/8/2015
Project 5: SMICRO Instruction Execution
Timing
CPC clock
RPC resetb
T
iadr
idat

<table>
<thead>
<tr>
<th>00</th>
<th>01</th>
<th>02</th>
<th>03</th>
<th>04</th>
<th>05</th>
<th>06</th>
<th>07</th>
<th>08</th>
<th>09</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>00</td>
<td>64</td>
<td>09</td>
<td>150</td>
<td>05</td>
<td>64</td>
<td>08</td>
<td>150</td>
<td>07</td>
<td>96</td>
<td>05</td>
<td>96</td>
<td>07</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>96</td>
<td>05</td>
</tr>
</tbody>
</table>

---

clock
resetb
CPC (Clock for Program Counter) = clock
RPC (Reset for Program Counter) = resetb
T (instruction clock) <= clock/2
iadr (instruction/program address)
idat (instruction op-code and operand)
### Project 5: SMICRO Instruction Execution Timing

The timing diagram illustrates the interaction between various components of the Simple Microcontroller (SMC), including the Control Unit (CU), Instruction Decoder (ID), Data Path (DP), and Memory Interface (MI). The diagram shows the flow of operations such as instruction fetch, execution, and data transfer. Key components include:

- **CPC clock**: Represents the clock signal that synchronizes the operation of the microcontroller.
- **RPC resetb**: Reset signal for the microcontroller.
- **T**: Timing signal indicating the phase of the clock cycle.

#### Instruction Execution Flow

1. **Instruction Fetch**: The microcontroller fetches an instruction from memory.
2. **Instruction Decode**: The ID decodes the fetched instruction.
3. **Register Read**: The CU reads the required registers.
4. **Data Transfer**: Data is transferred between the DP and the MI.
5. **Address buses**: addr, idat are used to access memory and data, respectively.
6. **Control signals**: Various control signals manage the flow of data and operations.

The diagram is annotated with specific timing points and operations, emphasizing the synchronization and sequence of events during instruction execution. The timing is critical for ensuring the correct operation and efficiency of the microcontroller.

*Date: 10/8/2015*
Project 5: SMICRO Instruction Execution Timing

10/8/2015
CPC clock
RPC resetb
T
iapd
idat
RIR

clock
resetb
CPC (Clock for Program Counter) = clock
RPC (Reset for Program Counter) = resetb
T (instruction clock) <= clock/2
iapd (instruction/program address)
idat (instruction op-code and operand)
RIR (Reset for Instruction Register) = (resetb)'
WIR (Write signal for Instruction Register) = clock * T'
CPC (Clock for Program Counter) = clock
RPC (Reset for Program Counter) = resetb
T (instruction clock) <= clock/2
iadr (instruction/program address)
idat (instruction op-code and operand)
RIR (Reset for Instruction Register) = (resetb)'
WIR (Write signal for Instruction Register) = clock' * T'
IRO (Instruction Register Output)
WDR (Write signal for Data Register) = clock' * T * LD
DOE (Data register Output buffer Enable) = T * ST
Mdat (Memory data bus)
\( \text{read/writeb} = (\text{clock}' \times T \times ST)' \)
asb (Address Strobe) = (T * (LD + ST))'

10/8/2015

Project 5: SMICRO Instruction Execution Timing
Project 5: SMICRO Instruction Execution Timing
Project 5: SMICRO Instruction Execution Timing

10/8/2015
RIR (Reset for Instruction Register) = (resetb)’
WIR (Write signal for Instruction Register) = clock’ * T’
IRO (Instruction Register Output) = AMOD
SDM (Select Data Register Mux) = AMOD
**RIR (Reset for Instruction Register)**  
= (resetb)’

**WIR (Write signal for Instruction Register)**  
= clock’ * T’

**IRO (Instruction Register Output)**

**SDM (Select Data Register Mux)**  
= AMOD
RIR (Reset for Instruction Register) = (resetb)'
WIR (Write signal for Instruction Register) = clock' * T'
IRO (Instruction Register Output) = AMOD
SDM (Select Data Register Mux) = AMOD
WDR (Write signal for Data Register) = clock' * T * LD
RIR (Reset for Instruction Register) = (resetb)'
WIR (Write signal for Instruction Register) = clock' * T'
IRO (Instruction Register Output) = AMOD
SDM (Select Data Register Mux) = AMOD
WDR (Write signal for Data Register) = clock' * T * LD
RIR (Reset for Instruction Register) = (resetb)'
WIR (Write signal for Instruction Register) = clock' * T'
IRO (Instruction Register Output) = AMOD
SDM (Select Data Register Mux) = clock' * T * LD
WDR (Write signal for Data Register) = T * ST
DOE (Data register Output buffer Enable) = T * ST
**SDM (Select Data Register Mux)**  = AMOD  
**WDR (Write signal for Data Register)**  = clock' * T * LD  
**DOE (Data register Output buffer Enable)**  = T * ST  

**10/8/2015  Project 5: SMICRO Instruction Execution Timing**
SDM (Select Data Register Mux) = AMOD
WDR (Write signal for Data Register) = clock' * T * LD
DOE (Data register Output buffer Enable) = T * ST
Mdat (Memory data bus)

10/8/2015
Project 5: SMICRO Instruction Execution Timing
SDM (Select Data Register Mux) = AMOD
WDR (Write signal for Data Register) = clock' * T * LD
DOE (Data register Output buffer Enable) = T * ST
Mdat (Memory data bus)

10/8/2015
Project 5: SMICRO Instruction Execution Timing
SDM (Select Data Register Mux) = AMOD
WDR (Write signal for Data Register) = clock' * T * LD
DOE (Data register Output buffer Enable) = T * ST
Mdat (Memory data bus) = (clock' * T * ST)'
rw (read/writeb)
DOE (Data register Output buffer Enable) = T * ST
Mdat (Memory data bus) = (clock' * T * ST)'

rw (read/writeb)

10/8/2015

Project 5: SMICRO Instruction Execution Timing
DOE (Data register Output buffer Enable) = T * ST
Mdat (Memory data bus) = (clock' * T * ST)'
rw (read/writeb) = (T * (LD + ST))'
asb (Address Strobe)
Project 5: SMICRO Instruction Execution
Timing
Project 5  Instruction Decoder Solution

NOP
LD  #9
ST  5
LD  #8
ST  7
LD  5
LD  7

clock
resetb
CPC
RPC
T
iadr
idat
RIR
WIR
IRO
SDM
WDR
DOE
mdat
rw
asb

Clock for Program Counter = clock
Reset for Program Counter = resetb
Instruction clock <= clock/2
Instruction/program address
Instruction op-code and operand
Reset for Instruction Register = (resetb)'
Write signal for Instruction Register = clock' * T'
Instruction Register Output
Select Data Register Mux = AMOD
Write signal for Data Register = clock' * T * LD
Data register Output buffer Enable = T * ST
Memory data bus
Read/write = (clock' * T * ST)'
Address Strobe = (T * (LD + ST))'