Shantanu Dutta

Datapaths
Race Conditions in Sequential Circuits and Processor

Lecture Notes # 2

EECS 366: Computer Architecture
Race Condition

If in the same clock cycle, two or more signals change state, the logic produces a new output which then is again written in the enabled combinational logic in the same clock cycle. This gives rise to the potential of the loaded combinational logic (an enabled FF) and reads back to the input of the FF if the output of the combinational logic module goes through (or is else).

2. Alternatively, a race condition is said to occur in a sequential circuit, if its state changes more than once in 1 clock cycle.
Correct State Transition Using Level-Sensitive Latches
2 level-sensitive latches

Comb. Logic

Clk

Required transition for the darkened arrow becomes incorrect transition corresponding to the dashed arrow:

Incorrect State Transition or Race Condition Using Level-Sensitive Latches
No Race Conditions Using M-S or Edge-Triggered FFs
No Race Conditions  Using 2-phase clocking and level-sensitive latches

Correct transition for the darkened arrow irrespective of the relative speed of different excitation (next state) outputs:

\[ T_{gap} > T_{skew}, T_{2-1} \geq T_{p,ff} + T_{su} + T_{skew} \]

\[ T_{1-2} \geq T_{p,ff} + T_{p,logic} + T_{su} + T_{skew} \]
Execution of ADD r1 r1 r3: Single-cc Race Condition

Control signals: read r1, r3, write r1 valid

alu

r1 = 10
r3 = 7
o/p = 17

Clk

Reg. File (level-sensitive FFs)

Reg. File (level-sensitive FFs)

alu

r1 = 17
r3 = 7
o/p = 17

Clk

Level-sensitive latches are used in the register file.

2. Thus this sets up the potential for race conditions in this datapath if to the FFs/memory and combinational logic in a classic sec. ckt. model.

1. The register file and ALU datapath with read and write buses correspond

Race Condition in a Processor Datapath
Execution of ADD r1, r1, r3: Edge-Triggered Reg. File; No Single-cc Race Condition

Previous control signals invalid

Control signals: read r1, r3, write r1 valid

$\text{ALU}$

$\text{Reg. File (edge-triggered FFs)}$
Execution of ADD r1 r1 r3-Phase Clocked Reg. File: No Single-cc Race Condition

Previous control signals invalid

Control signals: read r1, r3, write r1 valid

r1 = 10
r3 = 7
o/p = 7
alu

r1 = 10
r3 = 7

Previous control signals invalid

r1 = 10
r3 = 7

Execution of ADD r1 r1 r3

Reg. File (2-phase clocking)

alu

Control signals: read r1, r3, write r1 valid

r1 = 10
r3 = 7
o/p = 7
alu

r1 = 10
r3 = 7

Previous control signals invalid

r1 = 10
r3 = 7

Execution of ADD r1 r1 r3

Reg. File (2-phase clocking)

alu

r1 = 10
r3 = 7
o/p = 7
alu

r1 = 10
r3 = 7

Previous control signals invalid

r1 = 10
r3 = 7

Execution of ADD r1 r1 r3

Reg. File (2-phase clocking)
The last (n-th) cc. while the write control signals (write 1 in \text{MUL}_1, t^1, t^3) are valid only in read 1 in \text{MUL}_1, t^1, t^3 are valid for all n cc's that the operation takes.

4. In such cases, we need to ensure that the read control signals (read 1, t^1, t^3) are not sufficient to prevent race conditions, though they are necessary.

3. For operations that take multiple cc's (e.g., multiplication-division), one we have seen above.

2. If \( L = 1 \), then this broader race condition definition simplifies to the output can change more than once and be written to a register more than once during the time period of \( L \). This can cause the \( L \) operation during the duration of the operation. This can cause the operation of an operation that writes to a register, reads back to the input of the operation, or writes to an operation's writes to a register, feeds back to the input of the operation when the output of the last (n-th) cc.

| Multiplex Cycle Race Conditions |