I2CA 24 exercise 9

Wolfgang J. Paul

December 5, 2024

1 Speeding up the processor with MMU (20 Pt)

In user mode the sequence of execute and phase bits $(E, phase)$ is always

$$(0,0),(0,1),(1,0),(1,1)$$

Modify the design such that for instructions which are *not* loads or stores, i.e. for which the translation of the effective address is not necessary, the sequence is shortened to

$$(0,0),(0,1),(1,1)$$

Hint: you have to give the new definitions of Ein and *phasein*.

2 Timer Specification (20 Pt)

Define a hardware device with the following properties. The IO ports are

- a 64 bit $counter \in \mathbb{B}^{64}$
- a command and status register $cmsr \in \mathbb{B}^{32}$.

Two bits of *cmsr* are meaningful:

- $cmsr[0]$ ticking
- $cmsr[1]$ interrupt

While the timer is ticking the counter is decreased in every hardware cycle. The interrupt is activated, when the counter reaches zero. At this time the counter stops ticking. The interrupt can only be cleared by writing to *cmsr*

3 Timer Hardware (20 Pt)

Give a gate level design of

1. a timer
2. the memory system of a processor with a timer

4 Information side channels (20 Bonus Pt)

Consider a system with disk and timer. The time's interrupt signal triggers a bomb which destroys the disk. Now consider the following run: i) start disk access ii) start timer which generates an interrupt before the disk access ends. By our specification the disk is live and we can access it after the timer interrupt. But in reality the disk does not exist any more. So somewhere in our specification we must have excluded the possibility of a bomb. Where? Hint: think of 'embedded systems' and 'environment'

5 Detecting violation of software conditions (40 Pt)

Extend the pipelined processor design such that interrupt signals are generated when the following software conditions are violated

1. no branch or jump in delay slots
2. if an instruction writes to memory address x, then the next 3 instructions are note fetched from address x.

Do not try to design here the mechanism which rolls back the pipeline after an interrupt.