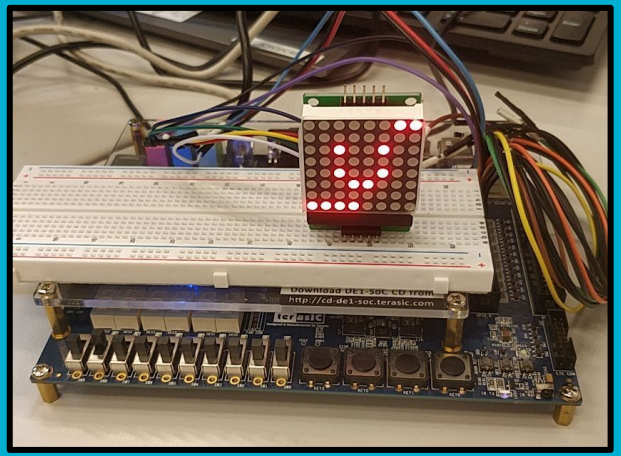
QuartusQuake

RTES - Project

Filip Slezak & Jakob Svedling

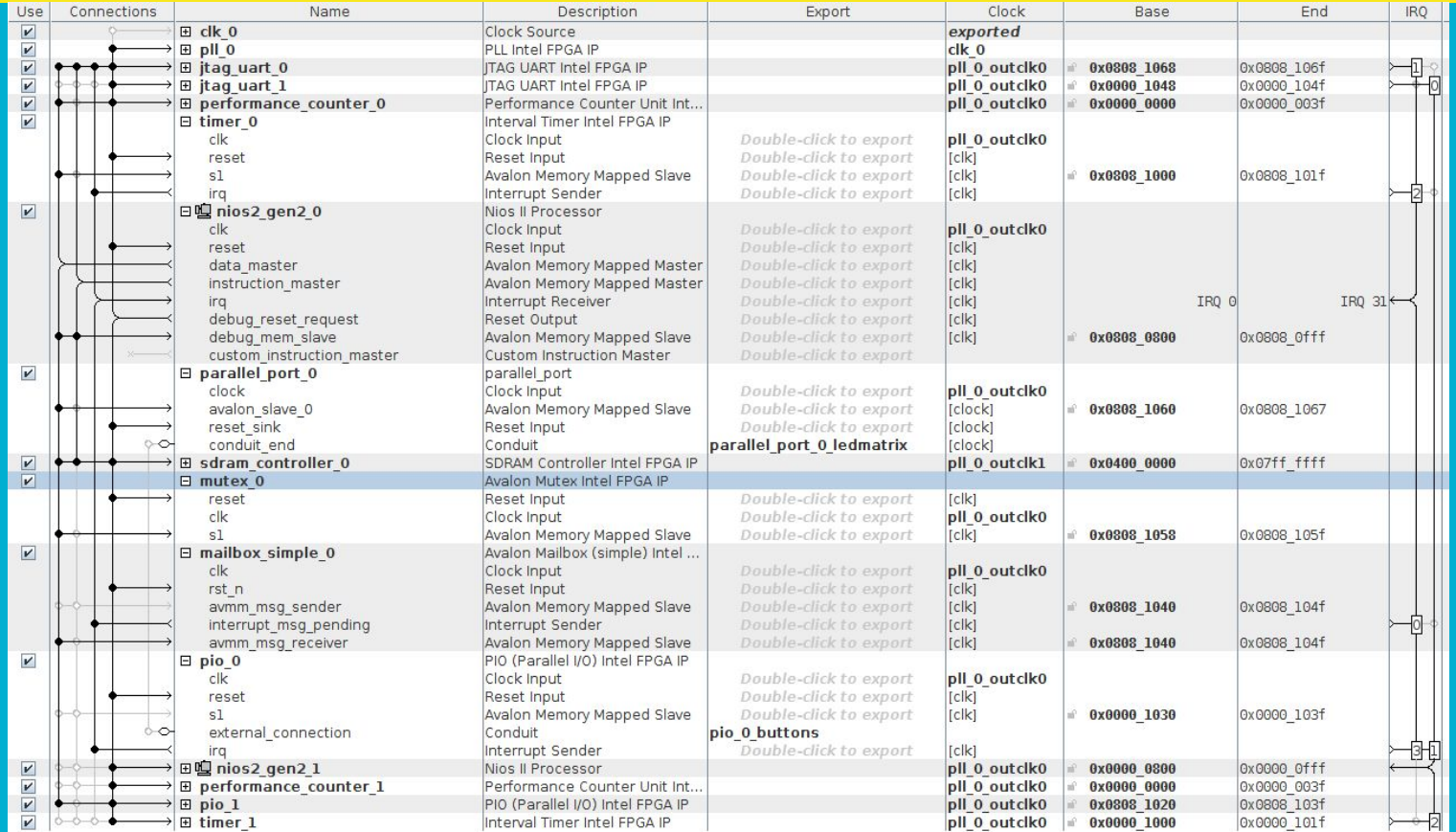
The Game & Setup

- QuartusQuake: a shooter minigame
- Input: Buttons (left/shoot/right)
- Output: Led Matrix
- Player vs Computer
- Goal: hit the opponent 8x to win
- Increasing difficulty



QuartusQuake Setup

Hardware Configuration



Use	Connections	Name	Description	Export	Clock	Base	End	IRQ
☑		⊞ clk_0	Clock Source		*exported*			
☑		⊞ pll_0	PLL Intel FPGA IP		clk_0			
☑		⊞ jtag_uart_0	JTAG UART Intel FPGA IP		pll_0_outclk0	0x0808_1068	0x0808_106f	1
☑		⊞ jtag_uart_1	JTAG UART Intel FPGA IP		pll_0_outclk0	0x0000_1048	0x0000_104f	0
☑		⊞ performance_counter_0	Performance Counter Unit Int...		pll_0_outclk0	0x0000_0000	0x0000_003f	
☑		⊟ timer_0	Interval Timer Intel FPGA IP					
		clk	Clock Input	Double-click to export	pll_0_outclk0			
		reset	Reset Input	Double-click to export	[clk]			
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0808_1000	0x0808_101f	
		irq	Interrupt Sender	Double-click to export	[clk]			2
☑		⊟ nios2_gen2_0	Nios II Processor					
		clk	Clock Input	Double-click to export	pll_0_outclk0			
		reset	Reset Input	Double-click to export	[clk]			
		data_master	Avalon Memory Mapped Master	Double-click to export	[clk]			
		instruction_master	Avalon Memory Mapped Master	Double-click to export	[clk]			
		irq	Interrupt Receiver	Double-click to export	[clk]		IRQ 0	IRQ 31
		debug_reset_request	Reset Output	Double-click to export	[clk]			
		debug_mem_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0808_0800	0x0808_0fff	
		custom_instruction_master	Custom Instruction Master	Double-click to export				
☑		⊟ parallel_port_0	parallel_port					
		clock	Clock Input	Double-click to export	pll_0_outclk0			
		avalon_slave_0	Avalon Memory Mapped Slave	Double-click to export	[clock]	0x0808_1060	0x0808_1067	
		reset_sink	Reset Input	Double-click to export	[clock]			
		conduit_end	Conduit	parallel_port_0_ledmatrix	[clock]			
☑		⊞ sdram_controller_0	SDRAM Controller Intel FPGA IP		pll_0_outclk1	0x0400_0000	0x07ff_ffff	
☑		⊟ mutex_0	Avalon Mutex Intel FPGA IP					
		reset	Reset Input	Double-click to export	[clk]			
		clk	Clock Input	Double-click to export	pll_0_outclk0			
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0808_1058	0x0808_105f	
☑		⊟ mailbox_simple_0	Avalon Mailbox (simple) Intel ...					
		clk	Clock Input	Double-click to export	pll_0_outclk0			
		rst_n	Reset Input	Double-click to export	[clk]			
		avmm_msg_sender	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0808_1040	0x0808_104f	
		interrupt_msg_pending	Interrupt Sender	Double-click to export	[clk]			0
		avmm_msg_receiver	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0808_1040	0x0808_104f	
☑		⊟ pio_0	PIO (Parallel I/O) Intel FPGA IP					
		clk	Clock Input	Double-click to export	pll_0_outclk0			
		reset	Reset Input	Double-click to export	[clk]			
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0000_1030	0x0000_103f	
		external_connection	Conduit	pio_0_buttons				
		irq	Interrupt Sender	Double-click to export	[clk]			3 1
☑		⊞ nios2_gen2_1	Nios II Processor		pll_0_outclk0	0x0000_0800	0x0000_0fff	
☑		⊞ performance_counter_1	Performance Counter Unit Int...		pll_0_outclk0	0x0000_0000	0x0000_003f	
☑		⊞ pio_1	PIO (Parallel I/O) Intel FPGA IP		pll_0_outclk0	0x0808_1020	0x0808_103f	
☑		⊞ timer_1	Interval Timer Intel FPGA IP		pll_0_outclk0	0x0000_1000	0x0000_101f	2

Software Task Division

CPU0 (C++)

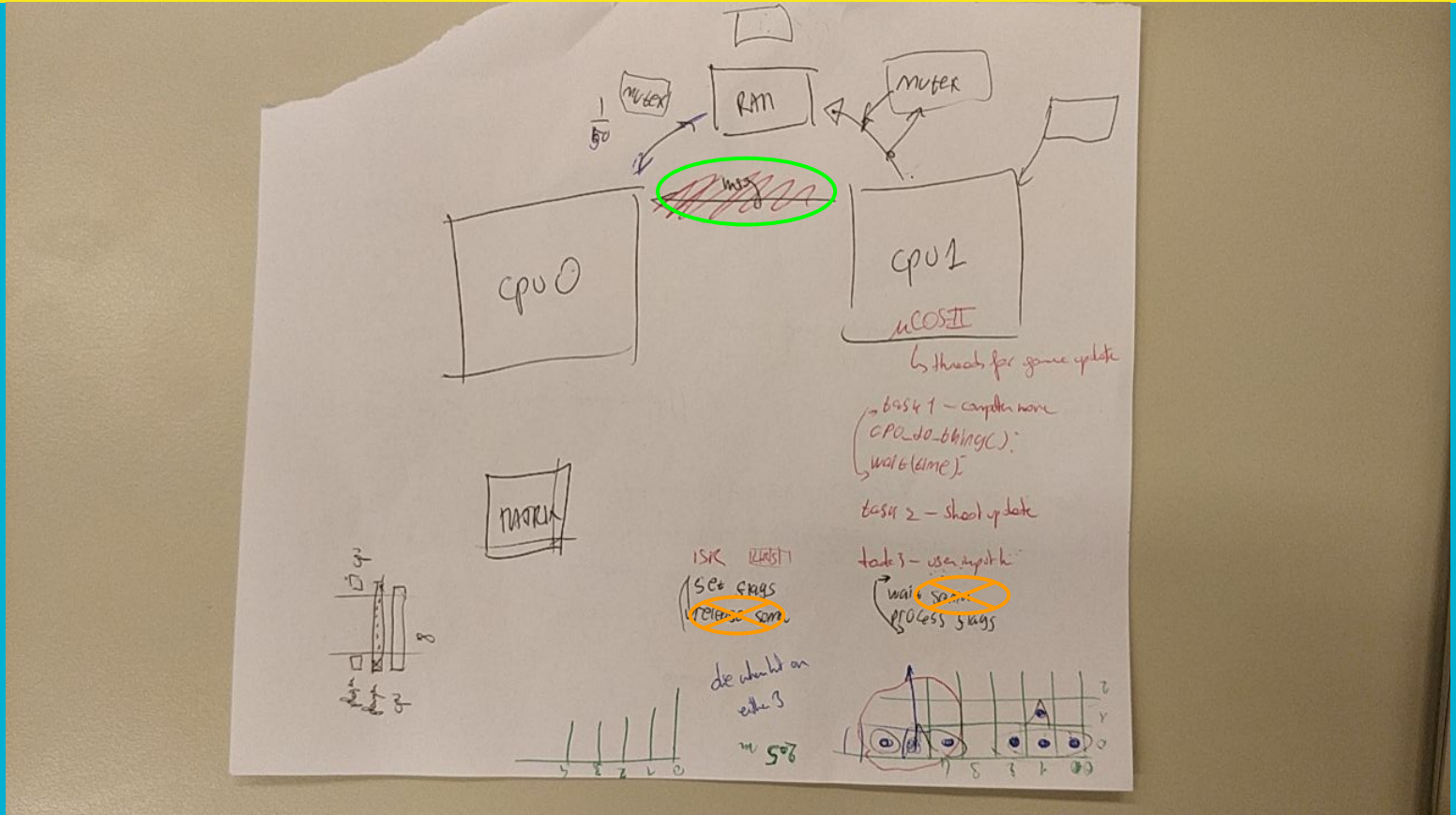
- Led matrix refresh
  - 50Hz Timer interrupt
  - SPI communication

CPU1 (MicroC/OS-II)

- Player moves
  - Button interrupts
- Computer moves
  - Aperiodic task ~ difficulty
- Game update
  - Aperiodic task ~ difficulty

- World map and game info on shared memory
- Initialization of pointers by mailbox message with address

Project Planning



Profiling Results

CPU0

Waiting for ISR to execute

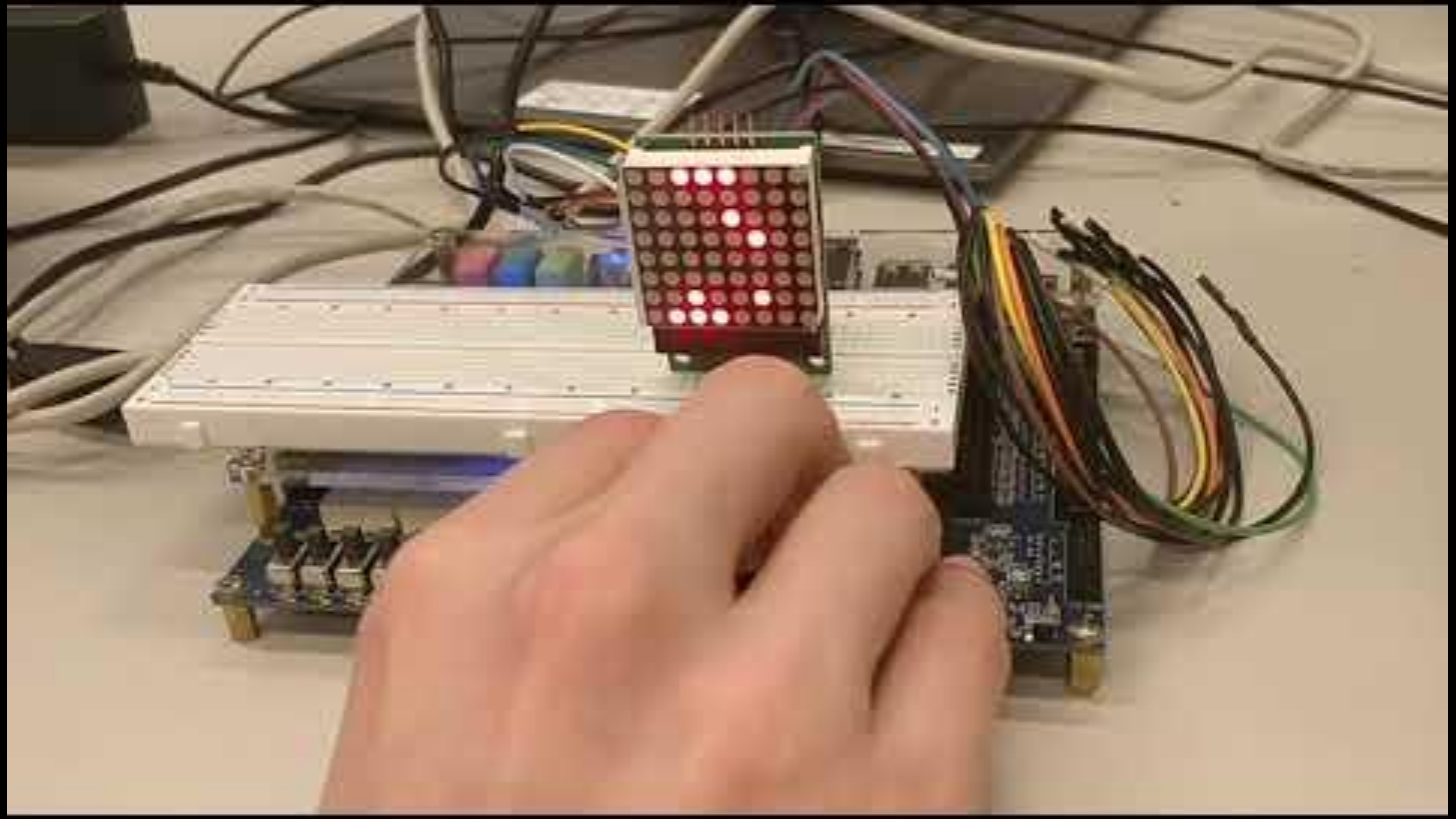
CPU1

Sleeps between tasks

CPU	Section	%	Time (s)	Occurrences	Avg. Time (ms)
CPU1	playermove	0.0523	0.0173	95	0.1825
	computermove	0.0778	0.0258	120	0.2152
	movebullets	0.2490	0.0827	180	0.4596
CPU0	ISR	60	19.7389	1618	12.1996

Table 1: The profiling results for Cpu0 and Cpu1 during a normal game from start to finish. (* % of the total processing time during execution of the profiler project design)

Demo Time



Lessons Learned (RTES Course)

- Familiar with embedded systems and FPGA
  - Discovered VHDL
  - Hardware speed up options
- Familiar with RTOS
- Better at reading technical documentation
- Much better at debugging
- Good reminder of C
- Unlocked new levels of patience