CSD
An introduction to digital systems using VHDL and to microcontroller applications using C

Table of content

This table of content is accompanied by references to each topic. Mainly, the references are examples, exercises, projects and class notes from digsys.upc.edu, but you can find slides and book chapters from all over the Internet. Be aware that for materials, which are not ours, the symbols, naming conventions and writing styles are different and thus, they must be renamed and reorganised in order to be in use in our CSD subject as valuable study materials.

1. Combinational circuits

1.1. Introduction to digital electronics

1.2. Logic gates

1.2.1. Circuits and logic diagrams

1.2.1.1. Buffer (non-inverter)

1.2.1.2. Inverter (NOT)

1.2.1.3. AND, NAND

1.2.1.4. OR, NOR

1.2.1.5. XOR, NXOR

1.2.1.6. Tri-state

1.2.2. Equations in Boole's Algebra

1.3. Truth table and canonical equations

1.3.1. Sum of minterms

1.3.2. Product of maxterms

1.3.3. Timing diagrams

1.4. Boolean functions

1.4.1. Properties

1.4.2. Duality principle

1.4.3. Product of sums (PoS)

1.4.4. Sum of products (SoP)

1.4.5. Any type of equation or non-standard forms

1.4.6. Wolfram Alpha engine for calculating truth tables and deducing logic circuits

1.5. Minimization of logic functions and circuits

1.5.1. Karnaugh Maps (not covered)

1.5.2. Espresso heuristic algorithm: minilog.exe, Logic Friday (not covered)

1.5.2.1. Minilog text format
1.5.2.2. Minilog minimisation table format
1.5.3. Don't-cares inputs ('x' or '-')
1.5.4. Incomplete functions: don't-cares outputs
1.6. Hardware description languages
1.6.1. VHDL
1.6.2. Verilog (not covered)
1.7. Commercial standard chips for logic gates
1.7.1. Logic families
1.7.1.1. TTL, LS-TTL, HC, HCT, etc. (not covered)
1.7.1.2. CMOS
1.7.2. Standard chip references
1.7.3. Electrical characteristics of logic gates
1.7.3.1. Voltage levels
1.7.3.2. Power dissipation
1.7.3.3. Propagation delay and maximum speed of operation
1.8. Virtual laboratory and simulation of electronic circuits
1.8.1. SPICE algorithms
1.8.1.1. Proteus ISIS from Labcenter Electronics
1.8.1.2. Multisim from National Instruments (not covered)
1.8.2. Digital simulators
1.8.2.1. Hades: interactive simulation framework (not covered)
1.8.2.2. Deeds: Digital electronics education and design suite (not covered)
1.8.3. Printed circuit board (PCB) design
1.8.3.1. Eagle (not covered)
1.8.3.2. Fritzing (not covered)
1.9. Designing combinational circuits using VHDL
1.9.1. Design flow using EDA tools
1.9.1.1. Specify
1.9.1.2. Plan
1.9.1.3. Develop (circuit synthesis)
1.9.1.4. Verification using a VHDL test bench
1.9.1.4.1. Functional simulation
1.9.1.4.2. Gate-level simulation
1.9.2. Plan A: single-file structural (equations)
1.9.2.1. Canonical equations: sum of minterms, or product of maxterms
1.9.2.2. Minimised equations: SoP or PoS (from minilog.exe)
1.9.2.3. Only-NOR gates
1.9.2.4. Only-NAND gates
1.9.2.5. Any kind of nonstandard equation
1.9.3. Plan B: single-file behavioural / high level /algorithm description
1.9.3.1. Translating the truth table
1.9.3.2. The truth table as a flowchart
1.9.4. Plan C1: Single-file hierarchical design (not recommended)
1.9.5. Plan C2: Hierarchical design (components and signals)

1.9.5.1. The method of decoders (MoD)
1.9.5.2. The method of multiplexers (MoM)
1.9.5.3. The method of ROM memory cells or RAM lookup tables (covered in Chapter 2)
1.9.5.4. Any combination of methods and equations
1.10. Standard combinational logic circuits
1.10.1. Design flow and EDA tools
1.10.1.1. Specifications
1.10.1.1.1. Symbol or entity, truth table, timing diagram, equations or algorithm
1.10.1.1.2. How to design larger multiplexers using smaller components of the same kind (plan C2)?
1.10.1.1.3. Commercial standard references
1.10.1.2. Plan A, B, C2
1.10.1.3. Development using EDA synthesis tools
1.10.1.4. Simulation using VHDL testbench
1.10.1.5. Gate-level simulation, propagation delay and maximum speed of computing
1.10.1.6. Laboratory prototyping using CPLD/FPGA training boards
1.10.2. Multiplexer: MUX_2, MUX_4, MUX_8, etc.
1.10.3. Demultiplexer: DeMUX_2, DeMUX_4, etc.
1.10.4. Encoders (priority high): Enc_3_8, Enc_4_16, etc.
1.10.5. Binary decoder: Dec_2_4, Dec_3_8, etc.
1.10.6. Hexadecimal to 7-segment decoder: HEX_7seg_decoder
1.10.7. Commercial standard logic chips in classic technologies
1.11. Binary codes and code converters
1.11.1. Radix-2 and radix-16 number systems
1.11.2. Gray
1.11.3. BCD
1.11.4. One-hot and one-cold
1.11.5. Johnson
1.11.6. ASCII, etc.
1.12. Binary numbers and arithmetic circuits
1.12.1. Addition
1.12.1.1. 1-bit adder (Adder_1bit)
1.12.1.2. *n*-bit adder (Adder_4bit, etc.)
1.12.1.2.1. Ripple-carry adder
1.12.1.2.2. Carry-lookahead adder
1.12.2. Ones counter
1.12.3. Multiplier (Mult_8bit)
1.12.4. Comparator
1.12.4.1. 1-bit comparator (Comp_1bit)
1.12.4.2. *n*-bit comparator (Comp_4bit, etc.)
1.12.5. Two's complement binary numbers (signed integers)
1.12.5.1. Subtractor
1.12.5.2. Adder-Subtractor (Int_Add_Subt_8bit, etc.)
1.12.5.3. Arithmetic Logic Unit
1.12.5.4. Multiplier (Int_Mult_8bit)

1.12.5.5. Comparator (Int_Comp_8_bit, etc.)
1.12.5.6. Parity generators and checkers
1.12.6. Commercial standard arithmetic chips in classic technologies
1.13. Programmable logic devices (PLD) and EDA tools for VHDL projects
1.13.1. Using a programmable array (PAL) to implement a logic function
1.13.2. Classic sPLD ispGAL22V10
1.13.3. Complex Programmable Logic Device (CPLD)
1.13.4. Field Programmable Gate Array (FPGA)
1.13.5. EDA tools for circuit synthesis
1.13.5.1. ispLEVER Classic / Diamond
1.13.5.2. Xilins Vivado / ISE
1.13.5.3. Intel Quartus Prime / Quartus II
1.13.6. VHDL simulation tools
1.13.6.1. VHDL testbench schematic for combinational circuits
1.13.6.2. ALDEC Active HDL Lattice Edition
1.13.6.3. Xilinx ISim
1.13.6.4. Mentor Graphics ModelSim Intel FPGA Starter Edition
1.14. Other examples and applications
1.14.1. Questionnaires
1.14.2. Problems
1.14.3. Preliminary laboratory assignments (PLA)

2. Sequential systems

2.1. The concept of memory in digital circuits. Current state and next state
2.2. Latch RS. Asynchronous 1-bit memory cell
2.2.1. Function table, state diagram, timing diagram
2.2.2. How to design a latch RS using gates
2.2.3. An RS latch circuit (RS_latch) using plan A
2.2.4. RS latch as a push-button debouncing circuit
2.3. Flip-flop. Synchronous 1-bit memory cell
2.3.1. The concept of a CLK signal. Synchronicity
2.3.2. CLK circuits
2.3.2.1. RC
2.3.2.2. Quartz crystal
2.3.2.3. Integrated circuit 555
2.3.3. The concept of clear direct (CD) or asynchronous reset
2.3.4. Power-ON reset and initialisation
2.3.5. Typical flip-flops
2.3.5.1. Deducing the circuit of an RS flip-flop from the RS latch, CLK's rising-edge detector
2.3.5.2. D, JK, and T flip-flops: symbol, state diagram, timing diagram
2.3.6. Analysis of asynchronous and synchronous circuits based on flip-flops and logic
2.4. Massive digital memories
2.4.1. Symbol, address and data
2.4.2. General architecture

2.4.3. ROM (read only memory)

2.4.3.1. VHDL circuit (ROM_2^*n*x*m*)

2.4.3.2. Method of ROM for implementing logic functions in VHDL, look-up tables (LUT)

2.4.4. RAM (random access memory)

2.4.4.1. VHDL circuit (RAM_2^*n*x*m*) and the idea of intellectual property (IP) specific circuits for a given target technology

2.4.4.2. Tri-state buffer in VHDL

2.5. Architecture of a Finite State Machine (FSM)

2.5.1. State register: *r*-bit memory (D_FF), state encoding (binary sequential, one-hot, etc.)

2.5.2. Next state logic (CC1), truth table, behavioural interpretation: flowchart

2.5.3. Output logic (CC2), truth table, behavioural interpretation: flowchart

2.5.4. Propagation delay from CLK to output (t_{CO}), maximum frequency of operation of a FSM

2.6. FSM in VHDL: a single-file project

2.6.1. VHDL testbench schematic for sequential systems: CLK and stimulus processes

2.6.2. Designing flip-flops as FSM

2.6.2.1. RS flip-flop (RS_FF)

2.6.2.2. D-type (data) flip-flop (D_FF)

2.6.2.3. JK flip-flop (JK_FF)

2.6.2.4. T-type (toggle) flip-flop (T_FF)

2.6.3. Examples of applications based on FSM

2.6.3.1. Classroom luminaries control

2.6.3.2. Push-button debouncing filter

2.6.3.3. Matrix keypad encoder

2.6.3.4. Traffic light controller

2.6.3.5. Stepper motor controller

2.7. Standard sequential systems as FSM

2.7.1. Binary counters

2.7.1.1. Symbol, function table, modulo, timing diagram, state diagram

2.7.1.2. Control signals: count enable (CE), up and down (UD_L) or reversibility, parallel load (LD) or output value preset

2.7.1.3. Outputs: binary code (binary sequential, BCD, one-hot, Gray, Johnson, etc.), terminal count (TC) pulse

2.7.1.4. Plan X: designing counters as FSM, single-file VHDL project (Counter_mod12, Counter_BCD_1digit)

2.7.1.5. Plan Y: designing counters using the VHDL arithmetic library and STD_LOGIC_VECTOR, single-file VHDL project (Counter_mod16)

2.7.1.6. Count truncation, expanding counters (Hour_counter)

2.7.1.7. Plan C2: designing counters using a hierarchical structure, standard components and logic, VHDL multiple-file project

2.7.2. Data register (Data_reg_*n*bits). Parallel load

2.7.3. Shift register (Shift_Data_reg_*n*bits. Shift left, shift right, parallel load

2.7.4. Commercial standard sequential chips in classic technologies

2.8. Dedicated processors

2.8.1. Architecture of an advanced digital system

2.8.2. Datapath (operational) unit, data input/output, status signals and flags

2.8.3. Control unit (FSM). Control signals, external inputs and outputs

2.8.4. CLK generator

2.8.4.1. Frequency divider

2.8.4.2. Pulsed to square waveform converter using T_FF

2.8.4.3. Adaptable generic structure

2.8.5. Examples and applications of dedicated processors

2.8.5.1. Serial multiplier

2.8.5.2. Serial adder

2.8.5.3. Serial asynchronous receiver and transmitter subsystem (SART)

2.8.5.4. Pulse generator

2.8.5.5. MM:SS timer

2.9. Other examples and applications

2.9.1. Questionnaires

2.9.2. Problems

2.9.3. Preliminary laboratory assignments (PLA)

3. Microcontrollers

3.1. Architecture and basics

3.1.1. Microprocessor

3.1.2. Microcontroller

3.1.3. Harvard – Von Neumann architectures

3.1.4. PIC18F4520 architecture

3.1.5. Assembler and C

3.2. MPLABX: Microchip Integrated Development environment (IDE) and XC8 C compiler

3.3. Digital inputs and outputs: example of a combinational circuit in C:

3.3.1. Hardware schematic

3.3.1.1. Polling switches and digital levels

3.3.1.2. CLK and reset circuits

3.3.1.3. Driving LED

3.3.2. Software program

3.3.2.1. I/O port configuration register (TRIS)

3.3.2.2. Program organisation: setup (*init_system*) and main loop (*read_inputs*, *truth_table*, *write_outputs*)

3.3.3. Development & testing

3.3.3.1. Project compilation and chip configuration files *hex*, *cof*

3.3.3.2. Proteus simulation and testing. Step by step debugging, watch variables window

3.3.4. Prototyping

3.4. Solving FSM in C

3.4.1. Interrupts. External event detection. Push-buttons

3.4.2. *output_logic*

3.4.3. *state_logic*

3.5. Peripherals: LCD

3.6. Peripherals: timer0,

3.7. Other examples and applications
3.7.1. Questionnaires
3.7.2. Problems
3.7.3. Preliminary laboratory assignments (PLA)