Electronic Technology Design and Workshop

Presented and updated by

Przemek Sekalski

DMCS room 2

2007

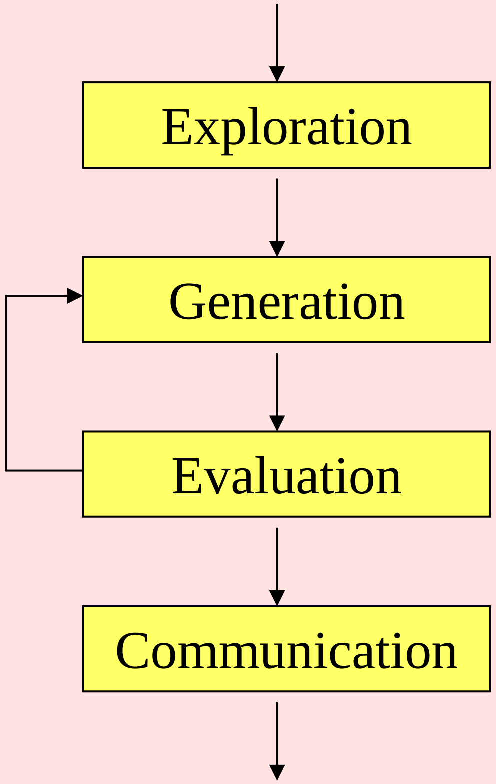
Electronic Technology Design and Workshop

Lecture 1

General concept

Design process methodology

example of a simple descriptive model of the design process



Necessity of *systematic* approach to the design process

Increasing complexity of modern designs - it is getting easy to overlook some construction details or constraints, which may have catastrophic consequences.

Need to develop team work - timing and coordination of the contribution of the team members, dividing the task into separate sub-problems and proper allocation of the team man-resources.

Mass production - avoiding (very!) high costs associated with defects detected after the mass production has progressed

Safety of large, advanced and unique projects - nuclear or chemical plants, space exploration, aircrafts etc., must be developed with very rigorous development algorithm.

Improved efficiency of the design process - minimise budget and time of new products.

Design process methodology

Exploration of Designs

At the start, the designer is faced with a very poorly defined problem - on the other hand the solution must be very well defined. So the task to accomplish is two-fold: understanding the problem and find the solution.

The ***problem and solution are complementary aspects of a design*** and must be developed side-by-side (designer makes a proposal that helps to understand what the problem ‘really’ is, and what proper solution should be.)

The exploration of design consist of generating the product ideas and analysing the solutions they might give.

Design process methodology

Generation of Designs

Generation of the first design proposal is the most creative part of designing. From the very brief statements of requirements, a designer proposes a draft of something new, never existing before (of course, most of the time, new products are modifications of the ones existing).

The most important feature for the designer is the ability to do ‘visualisation’ of the concept. At the early stage of the design process, very informal drawings help to visualise the concept, not yet to communicate with anyone else (a kind of thinking loud), and to help to preserve the instant ideas for future use.

The term ‘visualisation’ applies not only to real-life objects (like chair, car, building), but also to more abstract products (like electronic circuits, computer software), where the principles of operation or information flow diagram need to be ‘visible’ to the designer.

Design process methodology

Evaluation of Designs

The design, after being finished, must be evaluated for its completeness and feasibility.

- simple checks: whether parts fit together, assembly procedure, design rules, etc.,
- advanced checking: mathematical modelling, numerical analysis.

After errors are corrected or refinements applied, the design must be re-evaluated again.

Iterative process of refinement is a key part of the design process.

The potential danger is a loop of decision-making, where improvements in one part require changes to other part of the design and so on, delaying the design process (without guaranteeing better results).

Design process methodology

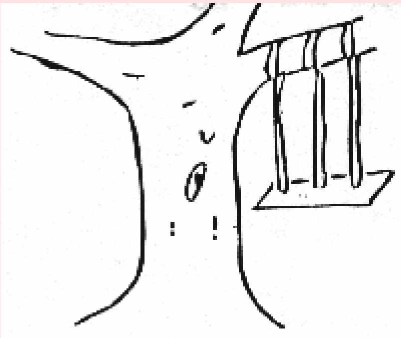
Communication of designs

The people who are making the product are not the same who designed it. The description of the new project must be communicated with all the most minute details and methods.

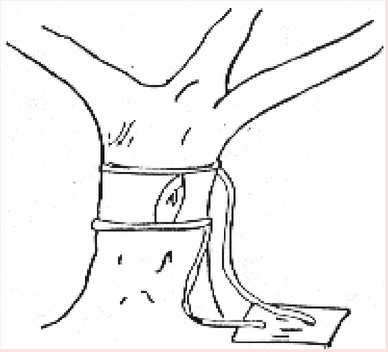
The design of a product is a documentation, i.e. the description of the manufacturing process (“a recipe for making”),usually in form of texts and drawings (shape, dimensions, materials, technology, etc.) or even prototypes, prepared in a formalised way.

Nowadays, it is not always a person who makes the product, but sophisticated machines with no direct human operator, so the final documentation may by in form of e.g. a string of numbers on a magnetic tape or a computer program.

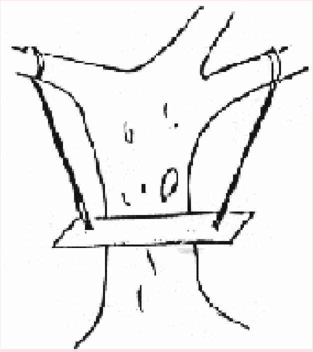
Design process methodology – the bad way



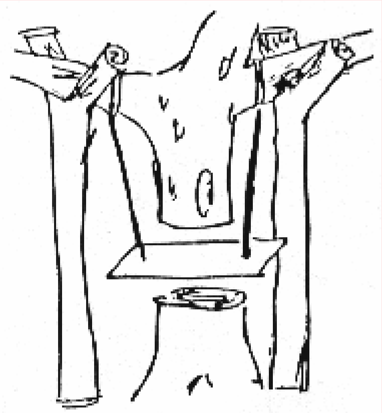
It is what the customer ordered



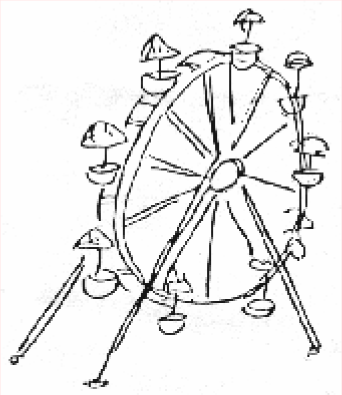
It is what was described in the project



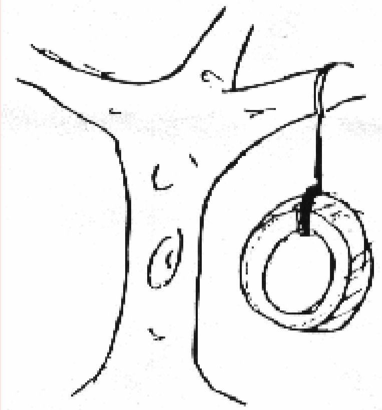
It is what was done by the engineers



It is what was done after commissioning



It is what the customer paid for



It is what the customer really need

Design methods

All systematic methods have two important features:
- ***formalisation*** - a way of avoiding the overlooking of details of the design
- ***externalisation*** - getting the ideas ‘out of the head’ of the designer into charts, diagrams, etc.

Methods of exploring design situations

- stating objectives
- literature searching
- interviewing users
- questionnaires
- investigating user behaviour

Methods of generating new ideas

- brainstorming
- removing mental blocks
- morphological charts

Methods of evaluation

- checklists
- selecting criteria
- ranking and weighting
- specification writing

Examples of design errors

HMS Sheffield radar system identified an incoming Argentinian Exocet missile as ‘non-Soviet’ and thus friendly. No alarm was raised. The ship was sunk with considerable loss of life.

A misplaced comma in one NASA program sent a Voyager spacecraft towards Mars instead of Venus.

Mariner 18 was lost due to missing NOT statement in program.

An F16 autopilot flipped the plane upside down whenever it crossed the equator. The fact that this occurred in a simulator doesn’t lessen the gravity of the mistake.

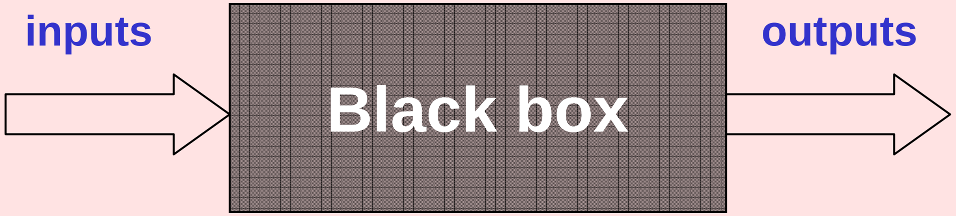
A 62 year old man died abruptly during treatment. Interference from the therapeutic microwaves had reset his pacemaker, driving his already injured heart to beat at 214 times a minute. It couldn’t do it.

A Shuttle laser experiment failed because the computer data was in nautical miles instead of feet.

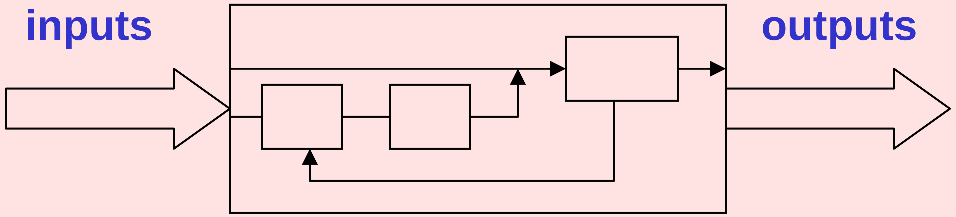
The Vancouver Stock Exchange Index rose by 50% when two years of round-off errors were corrected.

Electronic design representation

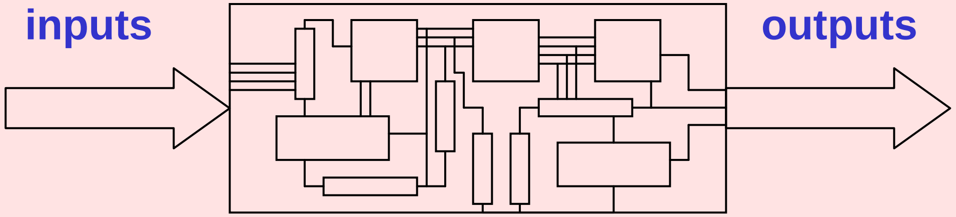
Behavioural representation



Structural representation



Physical representation



Electronic levels of abstraction

Design representation Abstraction level	*behavioural*	*structural*	*physical*
Transistor	Differential equations	Transistors, resistors, capacitors...	Analogue and digital cells (inside IC)
Gate	Boolean equations, state diagrams	Gate, flip-flop...	Modules, units (inside IC)
Register	Algorithms, flowcharts	Adders, comparators, registers, counters...	Microchips, IC's
Processor, IC	Programs, specifications	Processors, memories, controllers...	Printed circuit board (PCB)

Behavioural representation

Equations

$$I_{DS} = k_p \frac{W_{eff}}{L_{eff} - \Delta L}\left(U_{GS} - V_T - 0.5(1+F_B)U_{DSsat}\right)U_{DSsat}$$

$$F_B = \frac{\gamma F_S}{4\sqrt{2|\phi_f| - U_{BS}}} + F_N$$

$$\Delta L = X_D\left(\sqrt{\left(\frac{X_D E_p}{2}\right)^2 + \chi\left(U_{DS} - U_{DSsat}\right)} - \frac{X_D E_p}{2}\right)$$

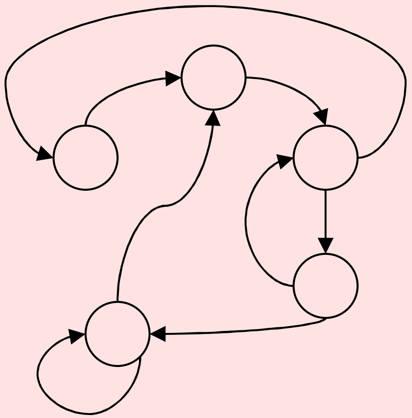
Boolean equations

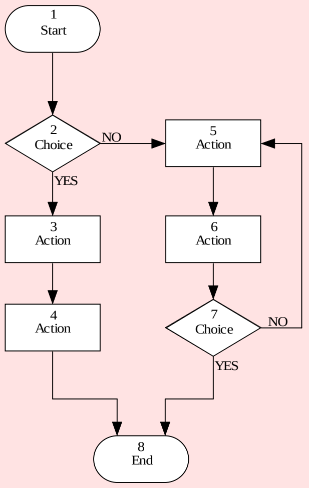
q1 := (!reset & !q3.FB & q1.FB & !q0.FB
  # !reset & !q3.FB & !q1.FB & q0.FB);

q1.C = (clock);

q1.OE = (!outen);

Flow charts & state diagrams



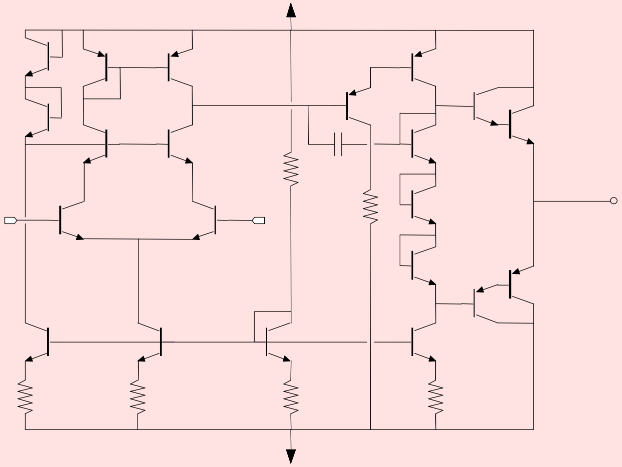


Programs

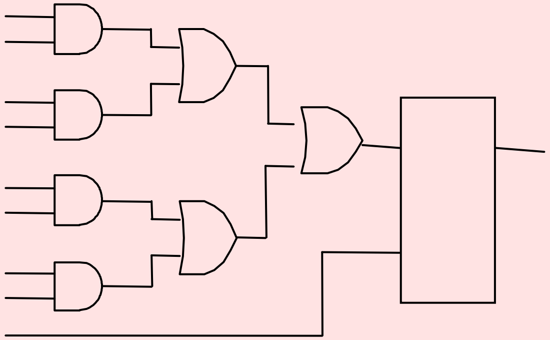
```
static loff_t vcs_lseek(struct file *file, loff_t offset, int orig)
{
        int size = vcs_size(file->f_dentry->d_inode);
        switch (orig) {
                default:
                        return -EINVAL;
                case 2:
                        offset += size;
                        break;
                case 1:
                        offset += file->f_pos;
                case 0:
                        break;
        }
        if (offset < 0 || offset > size)
                return -EINVAL;
        file->f_pos = offset;
        return file->f_pos;
}
```

Structural representation

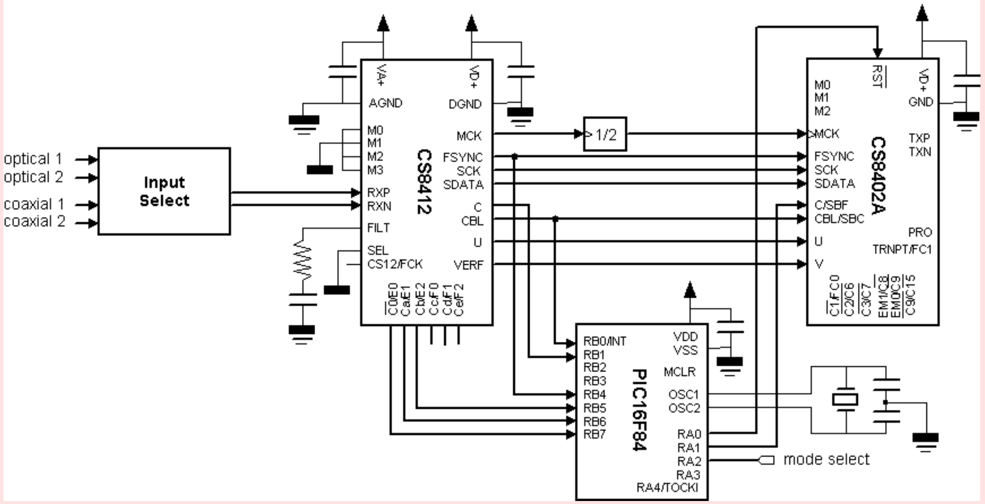
Transistors, resistors, capacitors...



Gates, flip-flops, registers

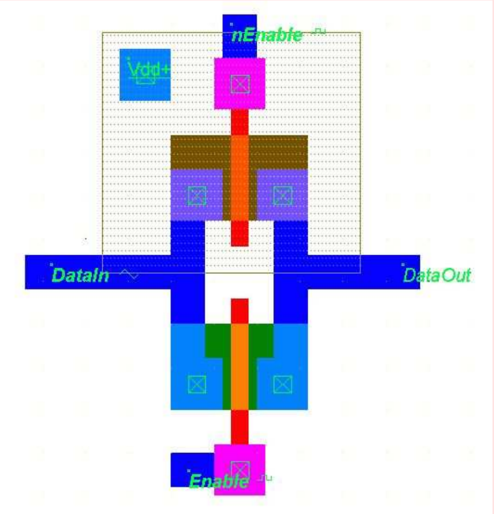


Processors, memories, controllers...

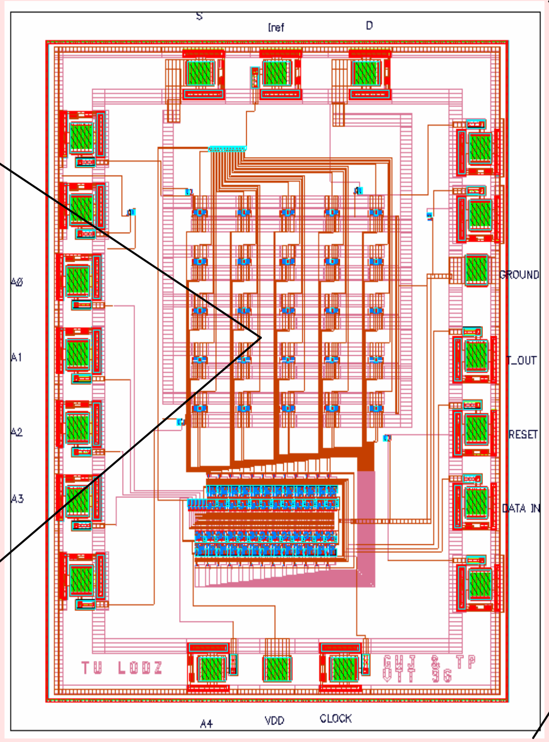


Physical representation

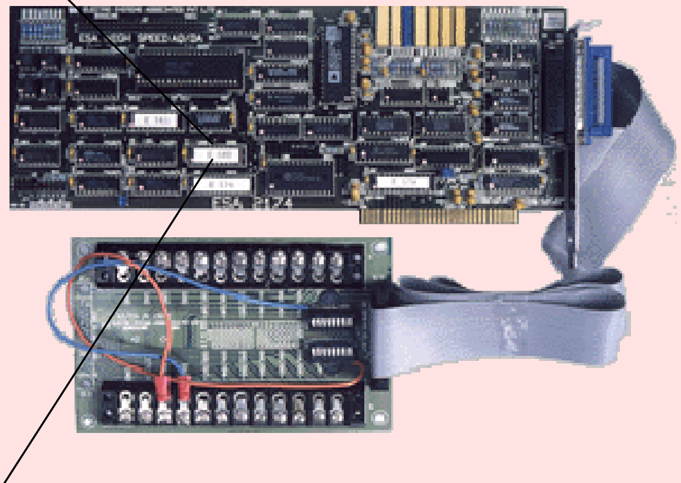
Element cells



IC layouts

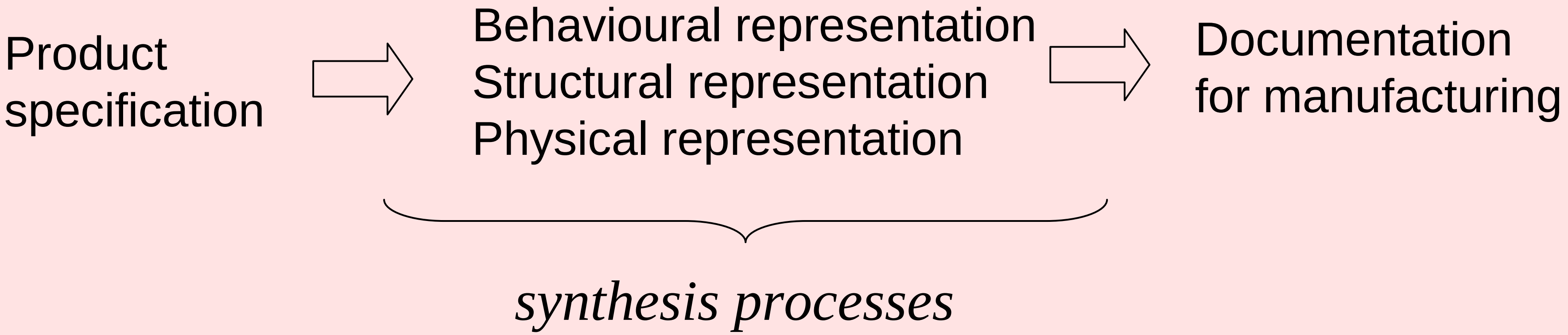


PCB layouts



Electronic design methodology

- level of abstraction
- synthesis tasks
- CAD tools



Product specification

Behavioural representation
Structural representation
Physical representation

Documentation for manufacturing

synthesis processes

Electronic synthesis

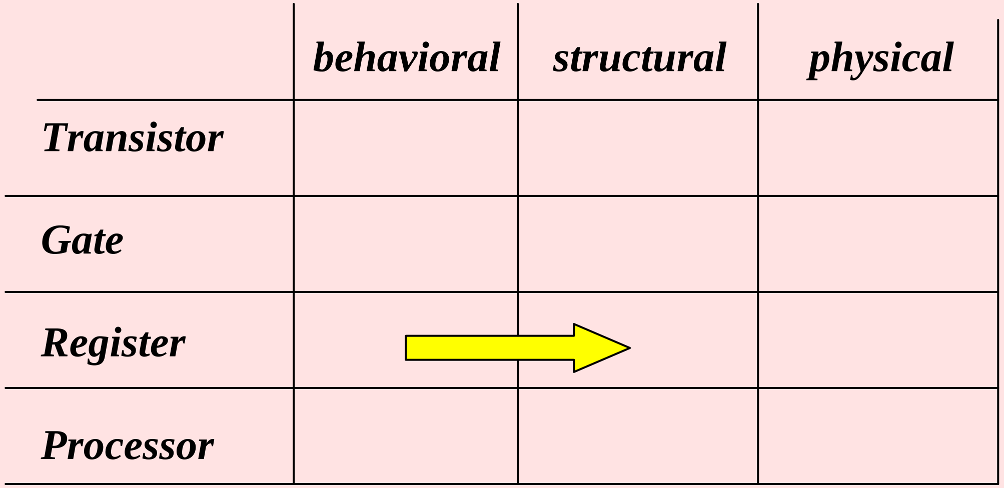
System synthesis - conversion of specification into a structure of processor level components

	behavioral	*structural*	*physical*
Transistor			
Gate			
Register			
Processor	→	→	

Process cannot be automated, creative human activity is necessary

Electronic synthesis

Architecture synthesis - conversion of algorithms and flow charts into a structure of register level components

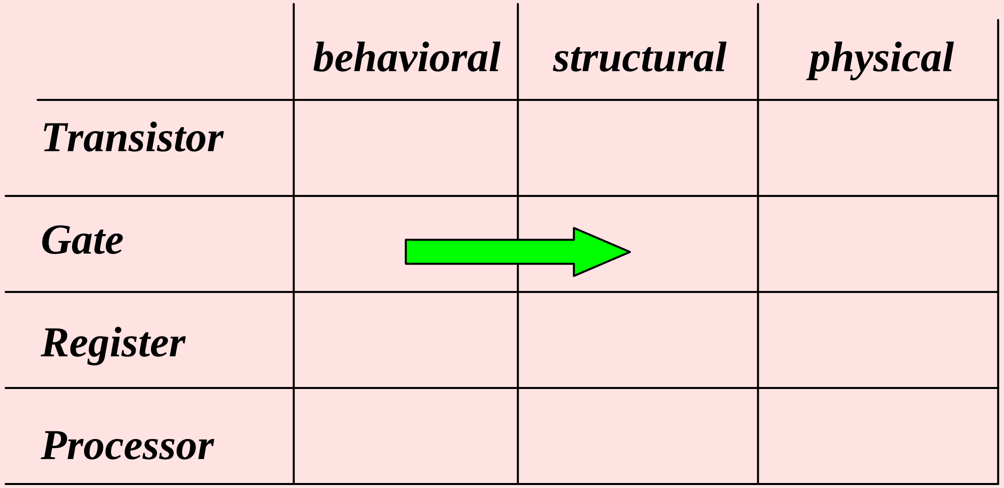


	behavioral	*structural*	*physical*
Transistor			
Gate			
Register	→		
Processor			

Process can be partially automated, creative human activity is helpful

Electronic synthesis

Sequential & Logic synthesis - conversion of state diagrams and boolean equations into gate level components



	behavioral	*structural*	*physical*
Transistor			
Gate	→	→	
Register			
Processor			

Process can be fully automated, creative human activity is not needed

Electronic synthesis

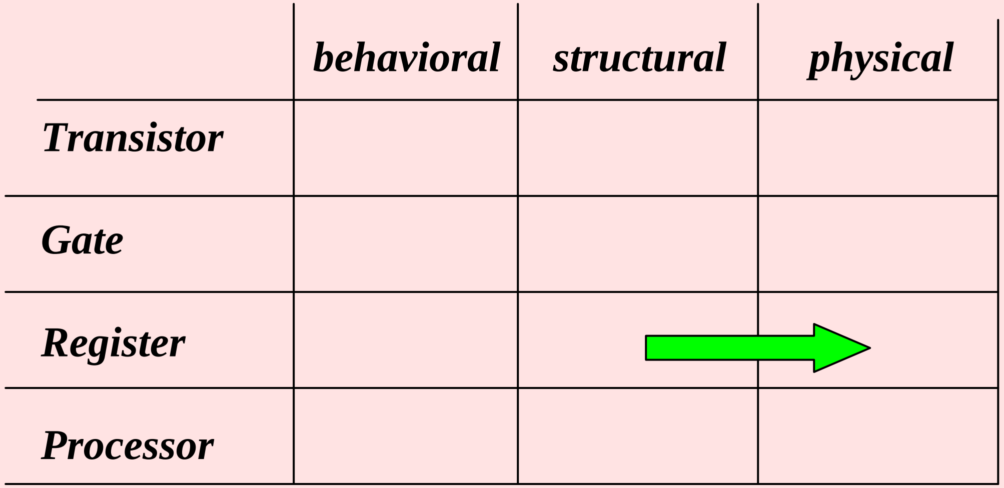
Layout synthesis - conversion of elements & gate level components into physical layout of the integrated circuit

	behavioral	*structural*	*physical*
Transistor		→	→
Gate		→	→
Register			
Processor			

Process can be fully automated, creative human activity is not needed

Electronic synthesis

PLD synthesis - conversion of register level components into an integrated circuit



	behavioral	*structural*	*physical*
Transistor			
Gate			
Register		→	
Processor			

Process can be fully automated, creative human activity is not needed

PLD programmable logic device

Electronic synthesis

PCB synthesis - conversion of processor level elements into physical layout of the printed circuit board

	behavioral	*structural*	*physical*
Transistor			
Gate			
Register			
Processor		→	

Process can be partially automated, creative human activity is helpful

PCB - Printed circuit board

CAD support in electronic design

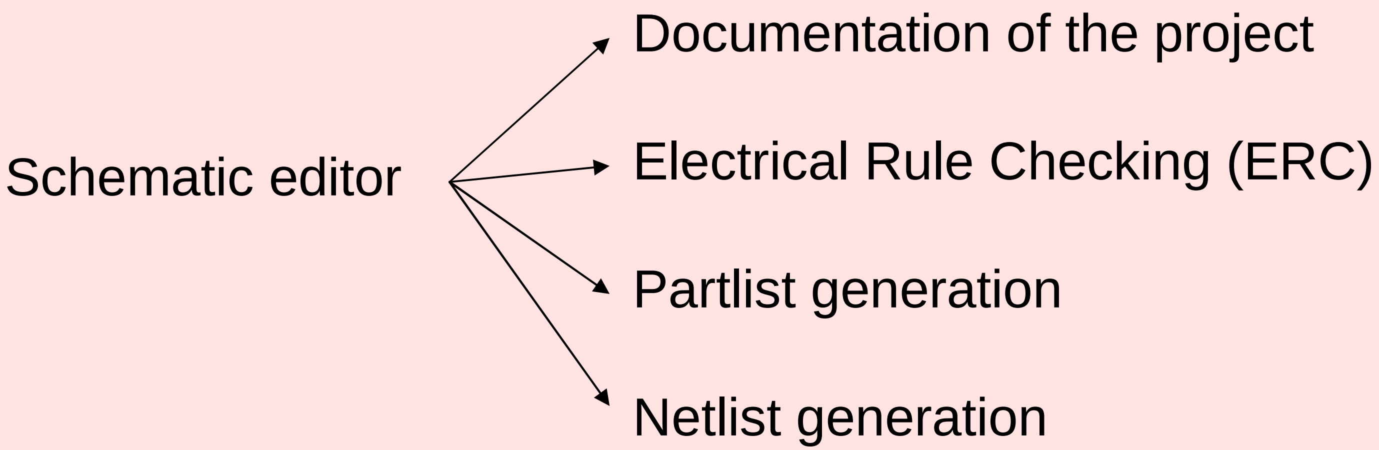
	behavioral	*structural*	*physical*
Transistor		🟩→	
Gate	🟩→	🟩→	
Register	🟨→	🟩→	
Processor	🟥→	🟨→	

🟥→ Not supported

🟨→ Partially supported

🟩→ Fully supported

CAD tools



Schematic editor

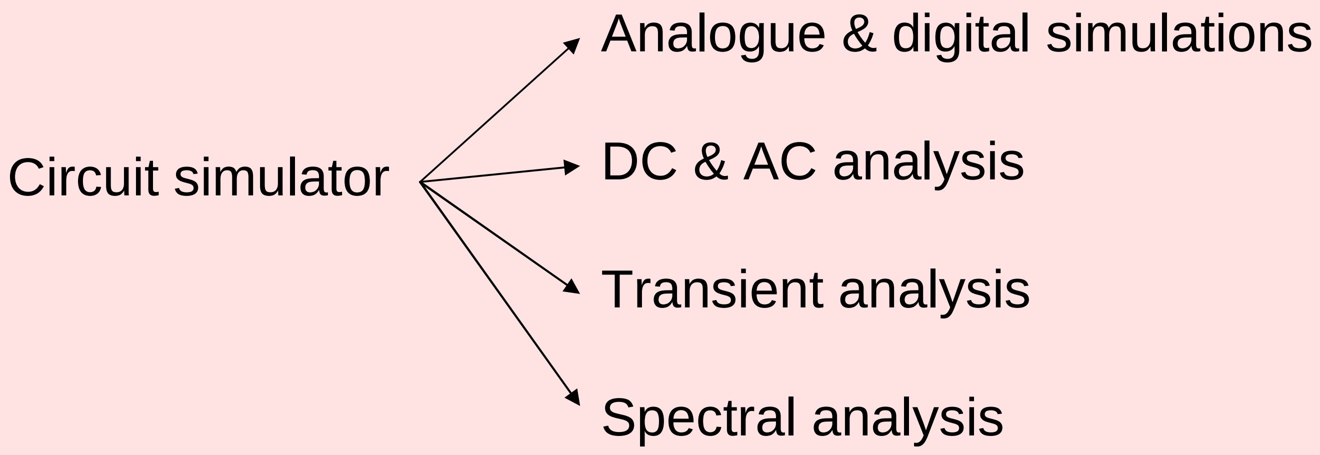
Documentation of the project

Electrical Rule Checking (ERC)

Partlist generation

Netlist generation

CAD tools



Circuit simulator

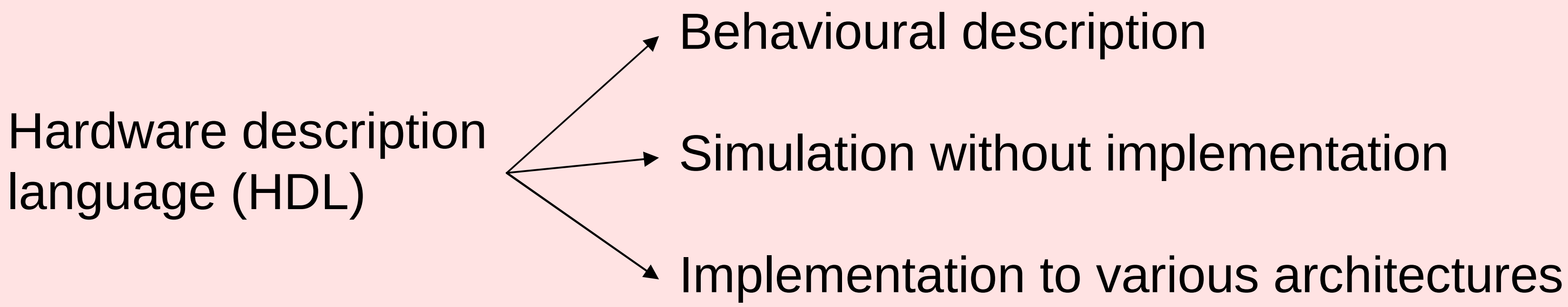
Analogue & digital simulations

DC & AC analysis

Transient analysis

Spectral analysis

CAD tools



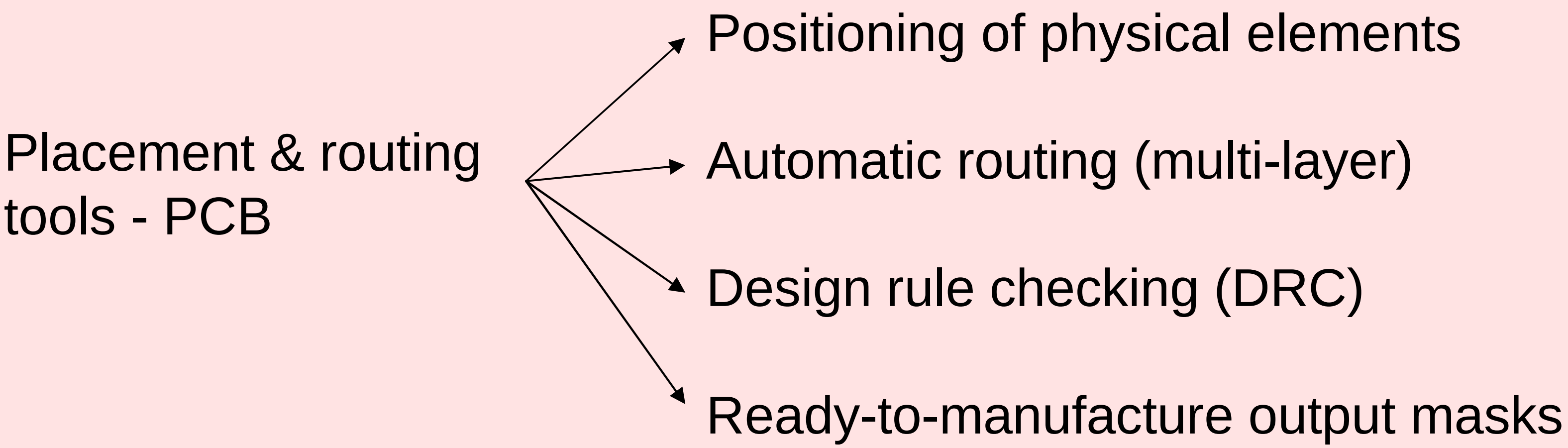
Hardware description language (HDL)

Behavioural description

Simulation without implementation

Implementation to various architectures

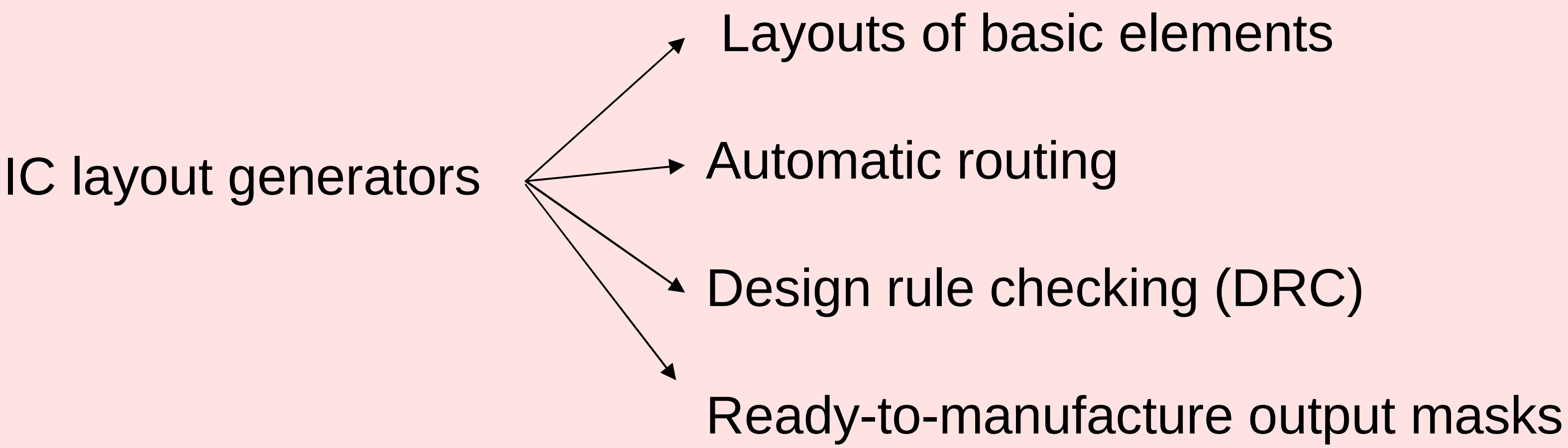
CAD tools



Placement & routing tools - PCB

- Positioning of physical elements
- Automatic routing (multi-layer)
- Design rule checking (DRC)
- Ready-to-manufacture output masks

CAD tools



IC layout generators

Layouts of basic elements

Automatic routing

Design rule checking (DRC)

Ready-to-manufacture output masks

ETDW course road map

✓ Schematic edition, libraries of elements

✓ Circuit simulation & netlist generation

✓ Microelectronics - full custom design and simulation

✓ Microelectronics - simple layout synthesis

✓ Hardware description languages - behavioural description

✓ Logic & sequential synthesis - programmable logic devices

✓ PCB design – auto-routing

🗐 Project - bringing the pieces together

Thank you for your attention