

### System-on-Chip Design for Wireless Communications

Stamenkovic, Zoran

Frankfurt (Oder), Germany, February 9-10, 2016 DFG-Workshop on Advanced Wireless Sensor Networks



innovations for high performance micro<sup>e</sup>lectronics



### Agenda

1	Wireless Systems (Hardware/Software Co-Design)
2	Modeling (Domains, Levels, and Basics of HDL)
3	Integrated Circuits (Analog, Digital, and Mixed)
4	Logic Synthesis (Low Power Techniques)
5	Technology and Layout (Low Power Techniques)
6	An Example (MAC Processor for IEEE Standard 802.15.3)

### Agenda



#### Wireless Systems (Hardware/Software Co-Design)

### **System Design Flow**



### **System Description – MAC Layer**

#### Data path

- Cyclic redundancy check (CRC) sum calculation
- Encryption and decryption of the frame payload
- Interfacing with the physical layer and frame buffering

#### Control path



#### Model the behaviour and performance of system design and architecture

- Develop a system concept and framework for implementation
- Verify that system may meet all requirements: performance, power consumption, cost, maintenance effort...

#### Model the performance of system implementation

- Detailed design and implementation of functional blocks and interfaces
- Verify robust behaviour in unexpected and erroneous situations
- Consider and compare different design styles (Software, Hardware)

#### System level design languages

UML, SystemC, SystemVerilog

### Hardware/Software Co-Design

#### Estimate performance for each block and compare with requirements

- Preferably using some dedicated high-level simulation tool for SLD language (SystemC, HandleC, or Matlab/Simulink)
- Tools having automatic code generator for HW and SW are desirable
  - Cadence CtoS (SystemC ⇒ Verilog)
  - Simulink HDL Coder (Proprietary Format ⇒ VHDL)

#### Criteria for implementing some block in HW or SW

- Required performance (data throughput, latency, ...)
- Functional complexity (normally easier in software)
- Expectations for later functional extension (easier in software)
- Complex or unusual arithmetic operations (easier in hardware)
   e.g. bit-related operations like in encryption, checksum / CRC
- Close relation to external interfaces (easier in hardware)

### **Functions Implemented in Hardware**

To retrieve Rx frame data from the physical layer byte by byte, perform filtering and CRC check, and store the data by means of direct memory access

To retrieve Tx frame data from a memory location, calculate and append the check sum, and push the data to the physical layer

To signal a successful reception or transmission of a frame by an interrupt

To analyse received and transmitted beacons and extract information on channel time allocations

To manage a queue of frames and select an appropriate frame for transmission

To perform the backoff procedure in the contention access period

To send an acknowledgment at the right time upon reception of a frame that needs to be acknowledged

To calculate the actual duration of a frame transmission based on its payload length and data rate

### **System Components**



www.ihp-microelectronics.com | © 2016 - All rights reserved

#### 2

#### Modeling (Domains, Levels, and Basics of HDL)

### **Domains of Modeling**



### **Levels of Modeling**



### **Hardware Description Languages**

#### Motivation for HDL

- Increased hardware complexity
- Design space exploration
- Inexpensive alternative to prototyping

#### **General features**

- Support for describing circuit connectivity
- High-level programming language support for describing behavior
- Support for timing information (constraints, etc.)
- Support for concurrency

#### VHDL

- IEEE Standard 1076-1987, 1993, 2000, 2002, 2008
- VHDL-AMS: Analog and mixed-signal extensions

#### Verilog

- IEEE Standard 1364-1995, 2001, 2005
- SystemVerilog

### **Design Methodology and Style**

#### Design methodology

- Size of system
- Design time constraints
- Cost and performance
- Available tools

#### Design style

- Full custom design (very tedious)
  - Most likely for data-paths widely used
  - Least likely for application specific circuits
- Semi custom or standard cell design
  - Application Specific Integrated Circuits (ASIC)
- Field Programmable Gate Array (FPGA) design
  - Prototyping oriented

### **Simulation and Test Bench**

#### Simulation

- Tests the functionality of a design's elaborated model
- Needs a test bench and a simulation tool
- Advances in discrete time steps

#### Test bench

- Includes an instance of the design under test
- Applies sequences of test values to inputs
- Monitors signal values on outputs using simulator

#### Simulation tools

- NCSIM (Cadence)
- VSIM (Mentor Graphics)
- VCS (Synopsys)

```
entity test_bench is
end;
architecture test_reg3 of test_bench is
    signal d0, d1, d2, en, clk, q0, q1, q2 : bit;
begin
    dut : entity work.reg3(behav)
        port map (d0, d1, d2, en, clk, q0, q1, q2);
    stimulus : process is
    begin
        d0 <= '1'; d1 <= '1'; d2 <= '1'; wait for 20 ns;
        en <= '0'; clk <= '0'; wait for 20 ns;
        en <= '1'; wait for 20 ns;
        clk <= '1'; wait for 20 ns;
        d0 <= '0'; d1 <= '0'; d2 <= '0'; wait for 20 ns;
        wait:
    end process stimulus;
```

end;



#### Integrated Circuits (Analog, Digital, and Mixed)

### **RF Integrated Circuits**



Phase Locked Loop

Voltage Control Oscillator



#### PLL 181 Osci core Core



### **Configurable General Purpose Processor**

Configurable processor cores can solve many processing tasks on SOCs and provide both software programmability and RTL processing speed

Small number of new, task-specific instructions can produce big performance gains with very small area increases

You need not to be a processor designer to use configurable processor cores for SOC designs



### **Digital Signal Processor**

#### DSP can have different data word sizes

Word size affects precision of fixed point numbers

Speed defined by speed of Multiply-Accumulate (MAC) operations

A DSP should keep the multipliers busy 100% of the time

#### Floating Point versus Fixed Point DSPs

- More expensive (2-4 times)
- Much slower

#### Basic DSP algorithms

- Infinite Impulse Response (IIR) Filters
- Finite Impulse Response (FIR) Filters
- Fast Fourier Transformers (FFT)
- Convolvers
- Turbo Decoders

### **Advanced Microcontroller Bus Architecture**



### Static Random-Access Memory (SRAM)

### Core cell uses six-transistor circuit to store value Value is stored symmetrically Both true and complement are stored on cross-coupled transistors SRAM retains value as long as power is applied to circuit Read

- Precharge bit and bit' high
- Set select line high
- One bit line will be pulled do

### Write

- Set bit/bit' to desired (complementary) values
- Set select line high
- Drive on bit lines will flip state if necessary



 $\Delta$ 



### **Memory Generators**

## A software tool which can create embedded memories (ROM or RAM blocks) in a range of sizes as needed

- The customer usually wants a particular number of words (depth) and bits (width) for each memory ordered
- Each of the final building blocks (physical layout) will be implemented as a standalone, densely packed, pitch-matched array

Complex layout generators and state-of-the-art logic and circuit design techniques offer

Embedded memories of extreme density and performance

#### Each memory generator is a set of various, parameterized generators

- Layout generator generates an array of custom, pitch-matched leaf cells
- Schematic generator and Net-lister extracts a net-list used for both layout vs. schematic and functional verification
- Function and Timing model generators create models for gate level simulation, dynamic/static timing analysis and synthesis
- Symbol generator generates schematic
- Critical Path generator is used for both circuit design and timing characterization

### Agenda

#### Logic Synthesis (Low Power Techniques)

### **Logic Synthesis**

#### Technology-independent optimizations

- A Boolean network is the main representation of the logic functions
  - Each node can be represented as sum-of-products (or product-of-sums)
  - Functions in the network need not correspond to logic gates

#### Technology mapping (library binding)

- Rewrites Boolean network in terms of available logic functions
- Optimizes for area and delay
- Finds the pattern match which minimizes area/delay cost
- Procedure
  - Write Boolean network in canonical NAND form
  - Write each library gate in canonical NAND form
  - Assign cost to each library gate
  - Use dynamic programming to select minimum-cost cover of network by library gates



### **Energy Consumption**

#### **Clock frequency**

■ f = 1/t

#### Energy

 $\blacksquare E = C_L (V_{DD} - V_{SS})^2$ 

#### Power

• E x f = f  $C_L (V_{DD} - V_{SS})^2$ 



#### Dynamic power dissipation comes from transistor switching

A single cycle requires one charge and one discharge of capacitor

#### Static power dissipation

Comes from leakage currents

#### Surprising result

- Resistance of the pull-up/pull-down transistor drops out of energy calculation
- Power dissipation is independent of the sizes of the pull-up and pull-down transistors

#### Circuit power-delay product (energy) is independent of frequency

Voltage and frequency scaling depend on this fact

### **Pipelining and Flattening**

#### Reduce critical path and glitches

 Savings due to supply voltage scaling or slower but more energy-efficient algorithms

#### Disadvantages

Area overhead with an implicit increase of capacitances



### Factorization



	Power dissipation components		
Implementation			
style	Total	Dynamic	Leakage
	power (µW)	power (µW)	power (nW)
imp1	3.26914	3.23945	29.6877
imp2	2.35462	2.33404	20.5844
imp3	1.87136	1.85511	16.2458



.....



### **Voltage and Frequency Scaling**

#### Architecture-driven supply voltage scaling

Add extra logic to increase parallelism so that system can run at lower frequency





#### Dynamic voltage and frequency scaling

- Decreased to parts of the circuit where it does not adversely affect the performance
- Dynamic scaling is regulated by software and based on system load

#### **Reducing capacitances**

- Parasitic capacitances of the transistors
- Parasitic capacitances of the wires

### **Clock Gating and Distribution**

#### Reducing switching activity

- Deactivate the clock to unused registers (clock gating)
- Deactivate signals if not used (signal gating)
- Deactivate V<sub>DD</sub> for unused hardware blocks (power gating)

#### Distributed clocks: Globally Asynchronous Locally Synchronous

- Eliminating centrally synchronous clocks and utilizing local clocks
- Distinct local clocks, possibly running at different frequencies



### Agenda

#### 5 Technology and Layout (Low Power Techniques)

### **Depleted SOI CMOS Manufacturing Technologies**



**Bulk CMOS** 



G

Partially Depleted SOI (PDSOI)



Fully Depleted SOI (FDSOI)

### **Tri-Gate FinFET**

#### Improved sub-threshold slope

Better control of short-channel effects



Drain

### **Power Gating**

#### Deactivation of the power supply for unused hardware blocks



### **Layout Generation**



To create a library database (standard cells, I/O cells, and macro blocks) with timing constraints

- Library Exchange Format (LEF)
- Timing Library Format (LIB, TLF)

#### To create a design database

Input Verilog net-list (V)

#### To create a chip layout database

 Output net-lists (V, DEF), timing (SDF), and layout (GDSII)

### **Layout Generation Flow**

#### Floor planning

Create a core area with rows (or columns) and I/O rows around the core area

#### Power planning and routing

Plan, modify and rout power paths, power rings and power stripes

#### Placement

- An I/O constraints file may be used to place the I/O pads
- Block placement
- Cell placement

#### Size adjustment

- Estimate the die size
- Resize design to make it routable

#### Generating clock trees

- Clock buffer space and clock net must be defined
- Generating clock trees is iterative process
- At this point, the physical net-list differ from the logical (original) net-list

#### Placement optimization

Resize gates and insert buffers to correct timing and electrical violations

#### Routing

Perform both global and final route on a placed design

#### Verification

Check for shorts and design rule violations

### Agenda

#### 6 An Example (MAC Processor for IEEE Standard 802.15.3)

### **SOC Design Flow**



#### SOC designs are a mix of

- Intellectual Property (IP) blocks
- Standard functions (inhouse developed)
- Application specific blocks (in-house developed)

# RTL descriptions, net-lists, and layouts

- Soft-cores
- Hard-cores
- UART, GPIO, PCMCIA
- AMBA, I<sup>2</sup>C bus
- Controllers
- Hardware accelerators
- SRAM and flash macros

### **Wireless Sensor Network Node**



#### IEEE 802.15.3 standard provides

- Ad-hoc networking, quality of service and security
- Various power management modes
- Physical layer data rates from 11 to 55 Mbit/s



## Thank you for your attention!

Stamenkovic, Zoran

IHP – Innovations for High Performance Microelectronics Im Technologiepark 25 15236 Frankfurt (Oder) Germany Phone: +49 (0) 335 5625 726 Fax: +49 (0) 335 5625 671 Email: stamenkovic@ihp-microelectronics.com

www.ihp-microelectronics.com



innovations for high performance microelectronics

