SoC++
Design Technology at
IMEC/DESICS

Diederik Verkest

http://www.imec.be/
http://www.imec.be/vsdm/

IMEC history

- Founded 1984
- Largest independent microelectronics research center in Europe
- Located in Leuven, Belgium
- Annual research budget 78 M US$
  - Personnel > 820 people
  - Close worldwide interaction with industry and with wide range of universities and research institutes
IMEC activities

- 21 415 m² facilities
  - 3 600 m² ultra clean processing area
  - 6 200 m² computer room & utilities
  - 10 400 m² offices & supporting laboratories
  - 1 215 m² training facilities

- R & D activities
  - Micro-electronics design and processing technologies
  - Balance between long-term and short- to medium-term research
  - Contract research including industrial residents, industrial affiliation program, transfer of qualified technologies

IMEC measures of success

- Scientific impact
- Cooperation with universities in complementary fields
- International network of cooperation with most of the important industrial performers in our field
- Portfolio of protected intellectual property
- Transfer of technologies to existing companies
- Creation of new spin-off companies
- Attracting foreign investments in the field of microelectronics and ICT
- Turn-over of well trained researchers to industry
IMEC activity areas

- DESICS: Design technology for integrated Information and Communication Systems
- Semiconductor Process Technology
- Silicon Technology and Device Integration
- Microsystems, Components, Packaging
- IC design training
- Associated University Laboratories

DESICS activities

- System-level design technology is inherently strongly linked to an application domain: communication and multimedia applications
  - Digital Broadband Transceivers
  - Wireless Systems
  - Multimedia Image Compression
- Total of appr. 120 people (incl. industrial residents, visiting researchers, PhD students, ...)

© IMEC 1999
Intelligent Home

- 155Mb/s WLAN
- 5GHz
- 10Gop/s <1 Watt
- MPEG 4 >100 Gop/s 5 Gtr/s 10 Watt

World’s first 155MB/sec WLAN

- Base station 155Mb/s multi-user rx antenna diversity
- Multi-path fading
- Orthogonal Frequency Division Multiplexing (OFDM)
- Turbo-coding
- Spatial Division Multiple Access (SDMA)
- Hiperlan-2/ IEEE 802.11 compatible
Single-package transceiver

- BiCMOS: RF circuitry
- CMOS: IF and digital circuitry
- MEMS: switches, varactor, resonators
- MCM: interconnect inductors, capacitors, resistors, filters, baluns

Multimedia MPEG-4 (member sctee)

- Diversity: 3D, Facial and Body Animation, Video
- Scalability: time, space, SNR
- Interactivity: behavior = f (input bits, user)
Design Technology Research

- Advances in processing technology
- Integration of PCB on single die: SoC

- Design productivity gap

SoC or …….. (S.O.S.)

- Design productivity gap grows!
  - Complexity increase 40% per year
  - Design productivity increase 15% per year
Design productivity gap

- System-level design
  - Concept to VHDL/C
  - Algorithms and architectures meeting performance, power, area constraints

- Physical design
  - VHDL/C to silicon
  - Timing closure (Monterey, Magma, Synopsys, Cadence, Avant! …)

System-level design

- Problems
  - 1 line of VHDL produces 40 transistors
  - $10^6 \text{tr/chip} = 250,000 \text{ lines VHDL code}$
  - Hardware AND (embedded) software
  - IP re-use

- Caused by
  - Abstraction level (VHDL ~ assembly)
  - Discrepancy design concepts and HDL semantics
Design concepts discrepancy

- HW designer
  - Signals
  - Clocks
  - FSMs
  - Data-paths
  - Instructions
  - I/F protocols
  - ...

- (V)HDL
  - Signals
  - Events
  - Processes
  - ...

How do I?

Design process

- Observations
  - Design is often incremental
    - Generations of the same product, re-use
  - Heterogeneous design environment
    - Difficult to navigate and explore design space
  - Informal system specifications
    - Impossible to back-track on system decisions
    - Ambiguous

- Solution
  - SoC++ object-oriented system design
System-level design

- Solution
  - Paradigm shift
  - Higher abstraction level
  - Object-oriented design
  - Re-use of HW (HDL) and SW (C/C++) compilers
  - Behavioral IP re-use
  - Incremental refinement to RT-HDL (HW) and C/C++ (SW)

SoC++ object-oriented design

- Eliminate conceptual gap between design concepts and language concepts
  - Design productivity increase
  - Enables behavioral re-use
- Design concepts = objects
  - Set of C++ object libraries implementing essential design concepts
  - Designers write down system-on-chip specification directly using these objects
SoC++ incremental refinement

- Introduce implementation detail in the same C++ framework
  - Floating-point to fixed-point
  - Data-flow to detailed cycle timing
  - Generation of register-transfer HDL code

- 922 lines HL-C++
- 4,426 lines RT C++
- 21,798 lines RT VHDL
- 154,952 lines gate level VHDL

SoC++ programming

SoC design = object-oriented programming activity
SoC++ programming env.

- **C++ object libraries**
  - Build object hierarchy on execution

- **C++ compiler**
  - Produces “executable specification” starting from any (mixed) abstraction level SoC description

- **Execute specification for**
  - Analysis: estimate performance, power, area, …
  - Verification: simulate to check functional correctness
  - Exploration: interactive refinement
  - Generation of low-level HDL/C code

Executable system spec.

- Executable “feasibility” phase
- Refinement of executable spec.
Historical perspective

- Design technology always lags processing technology

1971: Custom layout
1978: Symbolic layout
1985: Schematics capture
1992: HDL based design
1999: C/C++ based design

A SoC of the future

- Executable system spec.
- OCAPI
- Matisse
- TIPSY
- Mixed-Signal Design Technology
- FAST
- AFE
- ADSL
- DSP
- ATM proc.
- Micro-processor
- MPEG-4 co-processor
- FAST
- ATOMIUM
- PC/Windows
- PC/PC

© IMEC 1999
**Executable system spec.**

- **Requirements**
  - High simulation speed
  - Only capture relevant structure/function/timing detail
    - Maybe different from block to block
  - Performance estimation
    - Concurrency & Time

- **TIPSY fast functional verification with timing**
  - C++ class library
  - Multiple threads of execution
  - Simulated time for each thread
  - Guaranteed time consistency between threads
TIPSY - results for ADSL

Hardware (data-flow)

Software ("as is")

time-out specification → delay(200)
context switch → delay(2)

symbol rate 4kHz → delay(0.25)

TIPSY multi-threading & time

symbol rate 4kHz → delay(0.25)
time-out specification → delay(200)
context switch → delay(2)

- Fast timed co-simulation of HW and SW
- 10 seconds initialization ⇒ 30 minutes CPU
A SoC of the future

- Digital signal processing applications

OCAPI incremental refinement

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Data Type</th>
<th>Computational Model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Floating Point</td>
<td>Data flow</td>
</tr>
<tr>
<td></td>
<td>Fixed Point</td>
<td>RT</td>
</tr>
</tbody>
</table>

© IMEC 1999
OCAPI incremental refinement

OCAPI Design Experience

- 75 Kgate DECT modem
- VLIW architecture
- 22 data paths
- 18 MW (algor. -> gates)

- 80 Kgate Cable Modem
- DF architecture
- 26 FSMD
- 36 hrs (? in C++ -> gates)

Also: MPEG4 Wavelet Compressor, xDSL modem, OFDM modem, ...
A SoC of the future

- Network protocol processing applications
  - ATM, Internet Protocol (IP), TCP/IP, ...

Matisse incremental refinement

- Application domain
  - Large amount of data
  - Dynamically allocated
  - Real-time constraints

- Abstract Data Type refinement
  - Look-up tables, buffers, sets of timers

- Virtual Memory Management
  - Manage use of available memory
Matisse: ADT refinement

ATM_cell * Data_In;
Binary_Tree * RoutingTable;
Routing_Table = new Binary_Tree();
Data_In = new ATM_cell();
if ( Routing_Table->Lookup(Data_In) ) ...

Matisse: ADT results

- Library for incremental refinement
  - 28,355 lines of C++ code
  - Simulation/profiling
  - Implementation
- ATM multiplexer
  - previous
    - 327 mW, 601 mm²
  - new
    - 110 mW, 135 mm²
Matisse: memory management

- Concurrent OO spec
- Dynamic Memory Mgmt
- Task Concurrency Mgmt
- Physical Memory Mgmt
- Address Optimization

- SW design flow
- HW design flow

MATISSE - results

- Systematic exploration of dynamic memory management
  - Determine optimal power, area trade-off
  - Under guidance of designer

- Two steps
  - ADT refinement: determine optimal data-structures for abstract data types (storage + services)
  - VMM: * and & become addresses in static arrays

- Applied to SPP, F4, F5, ATMP, ...
A SoC of the future

- Multi-dimensional signal processing
  - Memory = performance bottleneck
  - Optimization of data transfers and data storage

ATOMIUM

- Memory = performance bottleneck
ATOMIUM exploration

- Memory = performance bottleneck
  - $P_{\text{off-chip mem. access}} = 30 \times P_{\text{arithmetic}}$
  - $P_{\text{on-chip mem}} = 40-60\% \times P_{\text{chip}}$
- Systematic exploration for reduction
  - Data storage cost
  - Data transfer cost
- Decrease power, area related to data transfer and storage
- Increase performance

ATOMIUM results

- Task Level DTSE
- Processor Level DTSE
- Optimization steps
- MPEG-4 Motion Estimation
- Resulting Power Reduction = 8
Multi-media compilers

- MPEG-4 on trimedia
- Factor 12 bus load reduction
- Factor 5 power reduction
- Factor 3.5 performance increase

SoC++: three messages to remember

- Executable system specification
  - TIPSY (towards an executable “feasibility” phase)
- Incremental refinement
  - Efficient path from executable spec to implementation under designer control
    - FAST (mixed-signal simulation)
    - OCAPI (physical layer of digital modems)
    - MATISSE (network protocol processing)
    - ATOMIUM (multi-dimensional “array” signal processing)
- Behavioral re-use
  - Re-use of functions *not* implementations
The DESICS pipeline

D6 RESEARCH PROGRAM

D6 / INDUSTRY TRANSFER PROJECTS

Industry product development

IMEC Industrial R&D Interaction

Training
evaluation & batch marking
contract research
industrial residentships
industrial affiliation program
licensing
information

© IMEC 1999
DESICS Industrial Affiliation Programs

MPEG-4 Applications
("White Box"-IP)

Advanced Wireless Local Communication Systems
("White Box"-IP)

Memory Optimization

Mixed Signal Design

IP Re-use

System Performance Modeling

HW/SW Co-design

HW/SW Partitioning

Training

Know-How Transfer

SOC++