# ASIPs as a Cornerstone of Heterogeneous MPSoCs: What, Why, and How?

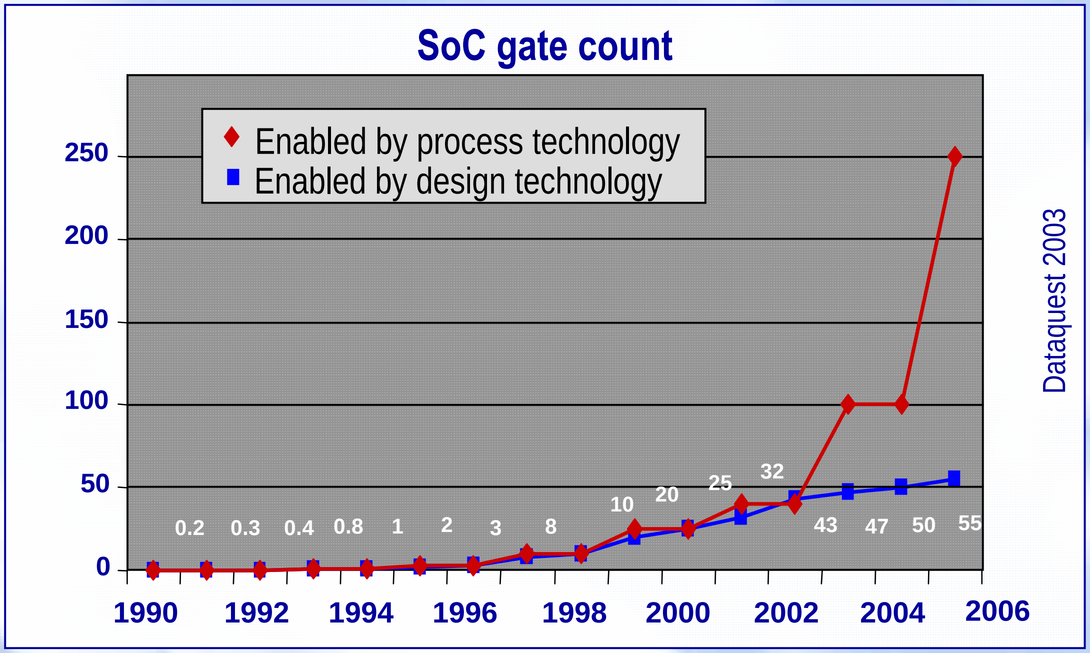
**Gert Goossens**
**Target Compiler Technologies**

gert.goossens@retarget.com
www.retarget.com

# Agenda

- SoC design trends
- Retargetable tool suite for ASIPs
- Multi-processor aspects
- Business model: IP or EDA?

# SoC design trends (1/3)

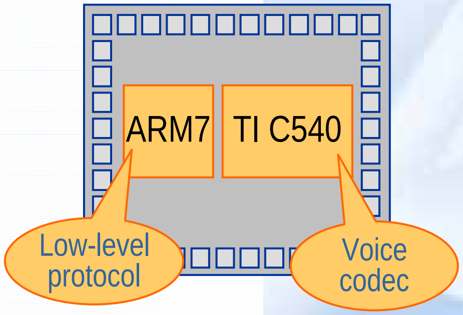


- **2003: Warning for looming complexity crisis [DataQuest 03]**
- **2006: How did industry respond?**
  - Multi-processor systems-on-chip
  - Important growth of programmable processor cores

# SoC design trends (2/3)

## System-on-Chip becomes Sea-of-Cores

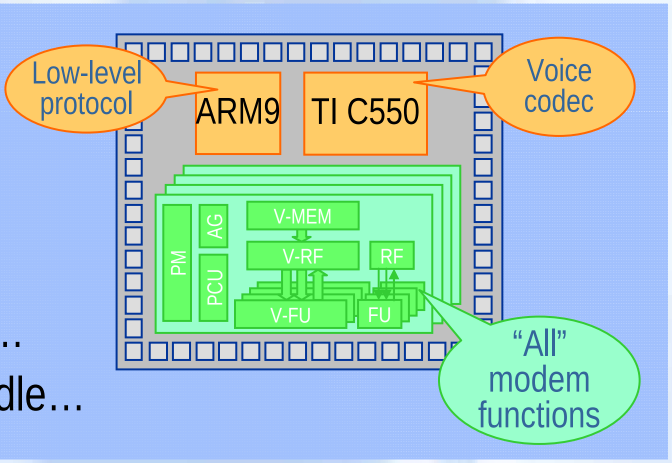
*2000*
**2G baseband chip**



*2005 ~ 2010*
**≥ 3G baseband & radio chip**
Multi-standard, SDR

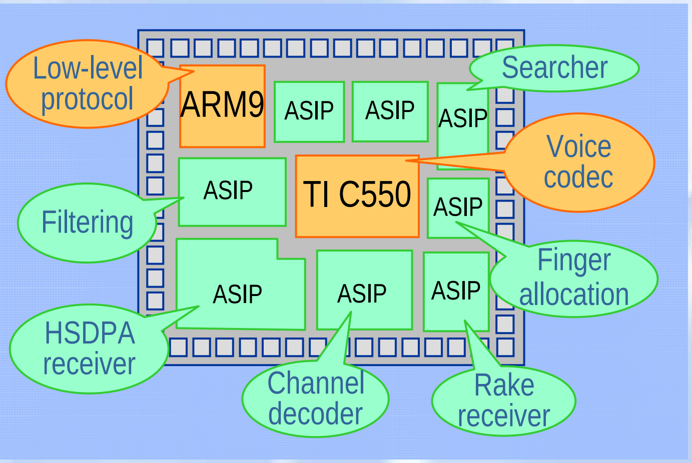
### Homogeneous MPSoC: general-purpose processors

- VLIW/SIMD: Philips/EVP, Sandbridge, OnDemand, Atmel/Diopsis…
- Array processor: Morpho, IMEC…
- Processor arrays: PicoChip, Cradle…



### Heterogeneous MPSoC: ASIPs

- Configurable IP vendors: Tensilica, ARC, Synfora, SiliconHive…
- **EDA vendors: Target**, CoWare…

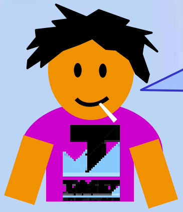


### Target focuses on heterogeneous MPSoC, but supports homogeneous as well

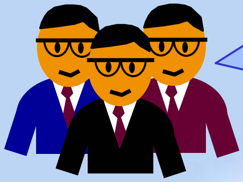
# SoC design trends (3/3)

## Example: HSDPA receiver

| | *ASIP-SIMD (Target customer)* | *General-purpose SIMD (EVP)* |
|---|---|---|
| *Technology* | 90 nm | 90 nm |
| *Data types* | 128 bit<br>*8x 16-bit int, 8x (8-bit, 8-bit) complex* | 256 bit<br>*16x 8 bit int, 16x (8-bit, 8-bit) complex* |
| *Clock* | 200 MHz | 300 MHz |
| *Gate count* | 250K | 450K |
| *Design time* | ~ 3 months | ? |

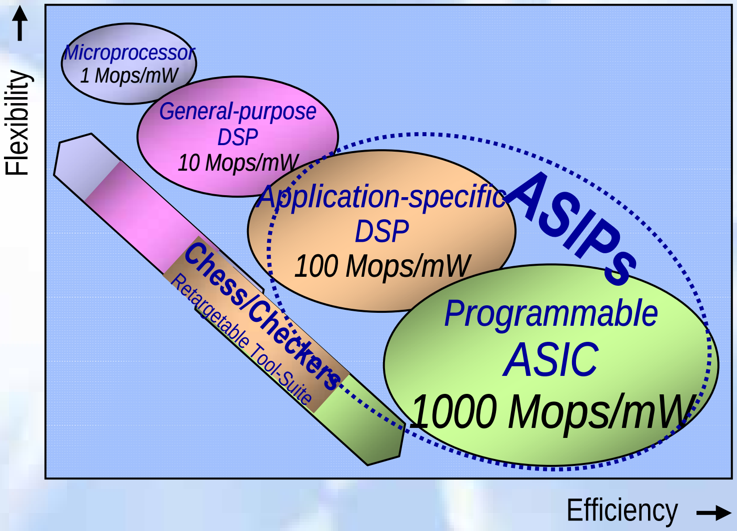


General-purpose solution may have its price: power, silicon real-estate



ASIP solution may have its price: design time, reliability, ease of programming

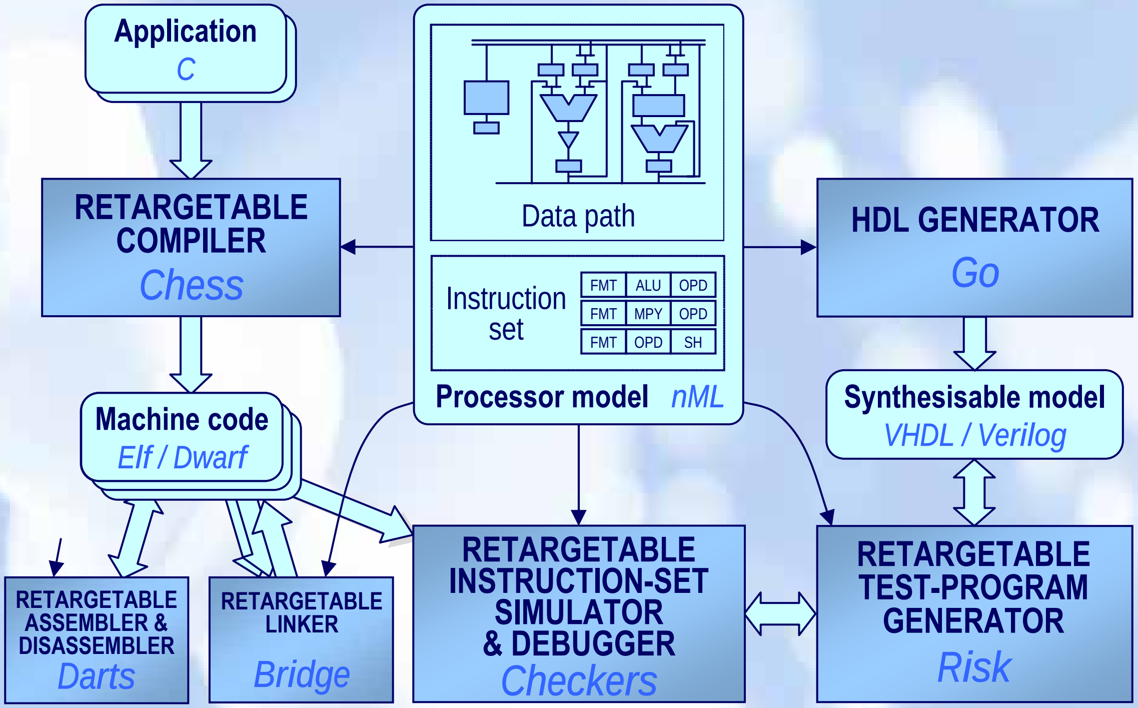
... unless supported by sound methodology and tools!!



# Agenda

- SoC design trends
- Retargetable tool suite for ASIPs
- Multi-processor aspects
- Business model: IP or EDA?

# Retargetable tool-suite for ASIPs (1/5)



Application
C

RETARGETABLE COMPILER
Chess

Data path

Instruction set

| FMT | ALU | OPD |
|---|---|---|
| FMT | MPY | OPD |
| FMT | OPD | SH |

Processor model nML

HDL GENERATOR
Go

Machine code
Elf / Dwarf

Synthesisable model
VHDL / Verilog

RETARGETABLE ASSEMBLER & DISASSEMBLER
Darts

RETARGETABLE LINKER
Bridge

RETARGETABLE INSTRUCTION-SET SIMULATOR & DEBUGGER
Checkers

RETARGETABLE TEST-PROGRAM GENERATOR
Risk

## Chess/Checkers tool suite supports:

- Architectural exploration and profiling
- HW generation and verification
- SW development : highly optimising C compiler, ISS, debug infrastructure
  → *single tool suite for all ASIPs* in heterogeneous MPSoC

# Retargetable tool suite for ASIPs (2/5)

❐ = multi-core design

🔒 = not disclosed

DSP →

| | |
|---|---|
| DSP for audio coding | *Philips* |
| DSP for audio coding | 🔒 |
| DSP for WLAN protocol code (VLIW) | *Philips* |
| DSP for fl-point num. processing (VLIW) ❐ | *Atmel* |

| | |
|---|---|
| ASIP for VDSL interference cancelation ❐ | 🔒 |
| ASIP for VDSL FFT ❐ | 🔒 |
| ASIP for ADSL FFT ❐ | *STMicro* |
| ASIP for ADSL equalisation ❐ | *STMicro* |
| ASIP for WLAN channel estimation ❐ | 🔒 |
| ASIP for WLAN mapper/demapper ❐ | 🔒 |
| ASIP for WCDMA rake receiver ❐ | 🔒 |
| ASIP for HSDPA receiver (SIMD) ❐ | 🔒 |
| ASIP for MPEG4/H264 video coding (SIMD) | *AOI* |
| ASIP for camera image processing | 🔒 |
| ASIP for audio coding ❐ | *Philips* |
| ASIP for GSM and VoIP speech coding | *Motorola* |
| ASIP for speech coding in cordless phones | *Sitel* |
| ASIP for hearing instrument audio ❐ | *Gennum* |
| ASIP for hearing instrument audio | 🔒 |
| ASIP for hearing instrument audio | 🔒 |
| ASIP for wireless comm in hearing instrument ❐ | *Philips* |

← Control

| | |
|---|---|
| Configurable microprocessors ❐ | *Easics* |
| OpenRisc 1200 | 🔒 |
| 8-bit microcontroller | 🔒 |

| | |
|---|---|
| ASIP for hearing instrument control ❐ | *Gennum* |
| ASIP for networking inter-process communication | *Motorola* |
| ASIP for 3G inter-process communication | *Freescale* |
| ASIP for packet parsing in DSLAM | 🔒 |

← General-purpose

Application-specific →

*Target is an interesting company in that they are reputed to have more design wins in [the ASIP tools] space than any of their competitors*

Clive "Max" Maxfield
[www.diycalculator.com/sp-compuniverse.shtml]

# Retargetable tool-suite for ASIPs (3/5)

### Arithmetic

- standard C
- any exotic functions

### Registers & interconnect

- general .. special purpose

### Instr. word

- orthogonal .. encoded
- fixed .. variable length

### Data types

- application-specific: integer, fractional, floating-point, bits, complex, vector…

### Instruction pipeline

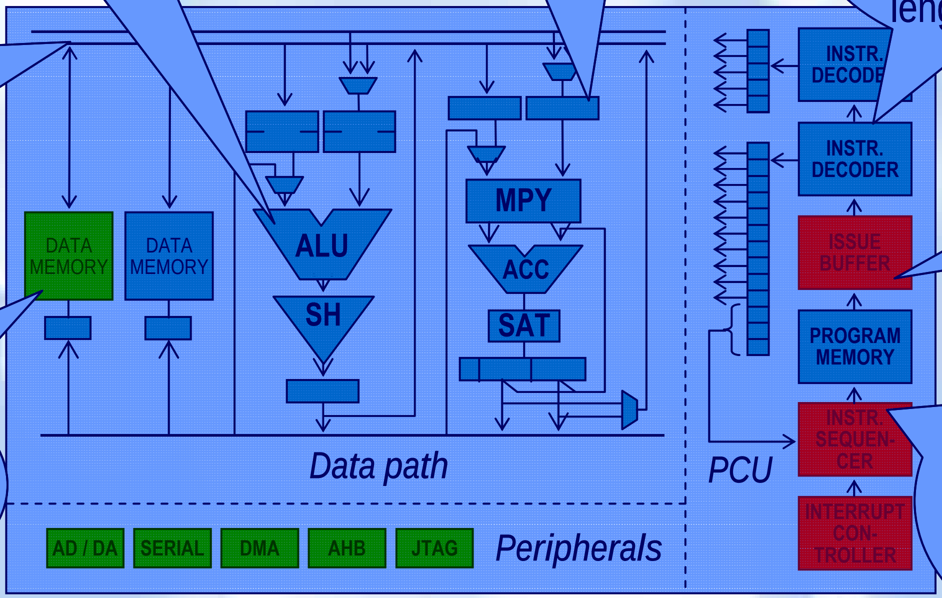
- any # stages
- pipeline hazards

### Memories & I/O

- any # memories & I/O ports
- varying latency
- addressing modes

### Prog. control

- subroutines
- interrupts
- HW do-loops
- residual control
- predication



## Broad architectural scope

- Optimise beyond the limitations of configurable templates
- True architectural exploration

# Retargetable tool-suite for ASIPs (4/5)

## nML : processor description language

- Programmer's model (cf. manual)
- Size: 1000–2000 lines — Learning: few weeks

```
mem DM[1024]<num,addr>;
reg R[4]<num>;
pipe C<num>;
trn A<num>; trn B<num>;
fu alu;
...
```

**Structural skeleton**

Storages & connectivity

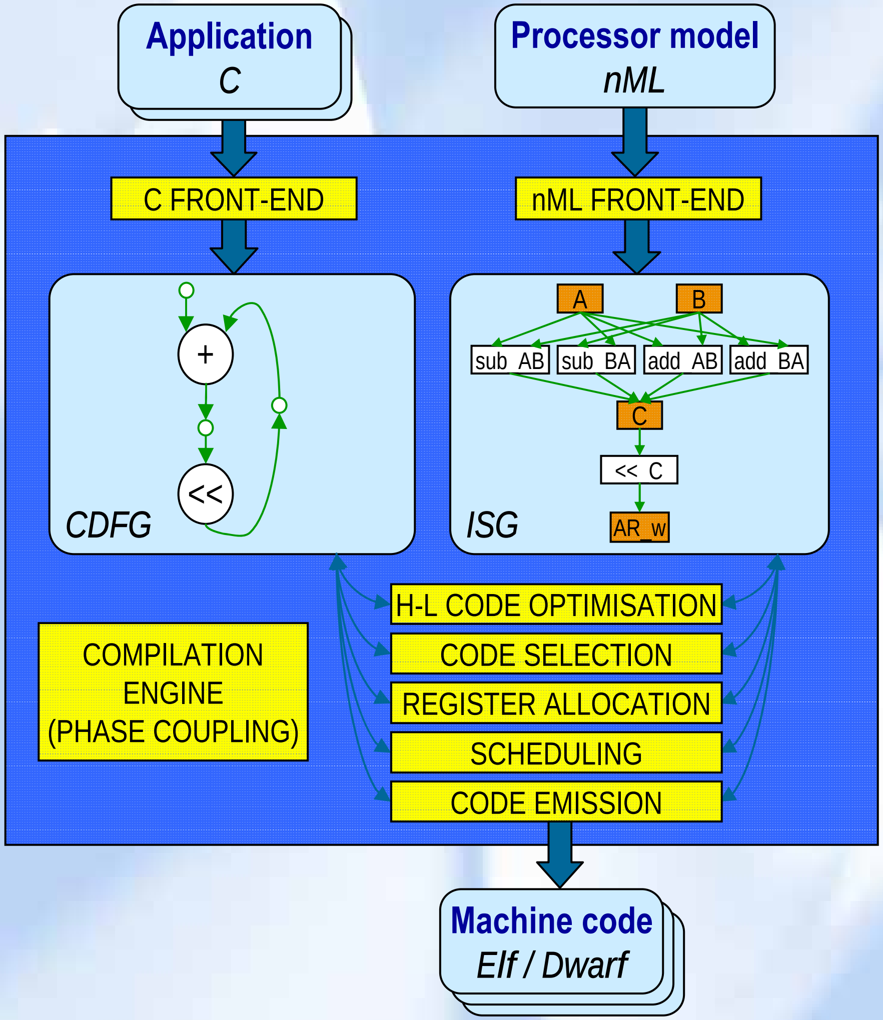
**Instr.-set grammar**

Instruction classes defined by register-transfer model

```
opn my_core (alu_inst | mac_inst
             | shift_inst);
...
opn alu_inst (op:opcod, x:c2u,
              val:c16s, y:c2u) {
  action {
    stage EX1:
      A = R[x];
      B = val;
      switch (op) {
      case add : C = add(A, B) @alu;
      case sub : C = sub(A, B) @alu;
      case and : C = and(A, B) @alu;
      case or : C = or(A, B) @alu;
      }
    stage EX2:
      R[y] = C @alu;
  }
  syntax : op " R" y ", R" x ", " val;
  image : "0"::op::x::y::val;
}
...
```

# Retargetable tool-suite for ASIPs (5/5)

## Chess : graph-based C compiler



### Front end

- C → Control-Data Flow Graph
- nML → Instruction-Set Graph

### Compilation phases

- Map CDFG onto ISG
- Graph algorithms

### ISG contains structural info

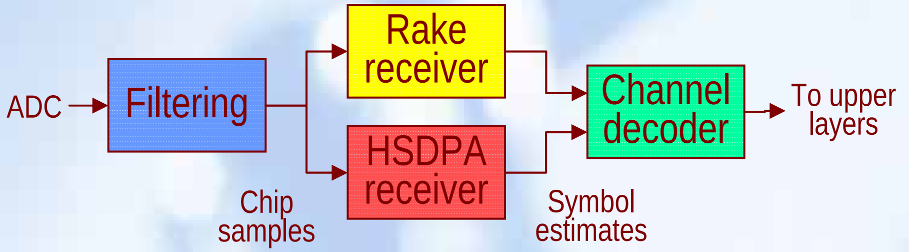
- E.g. hardware resources, data-types, connectivity, instruction encoding, pipelining, parallelism, pipeline hazards
- Much closer to hardware than conventional compilers (e.g. gcc)
- Enables efficient compilation for “irregular” architectures
- Patented

# Agenda

- SoC design trends
- Retargetable tool suite for ASIPs
- Multi-processor aspects
- Business model: IP or EDA?

# Multi-processor aspects (1/2)

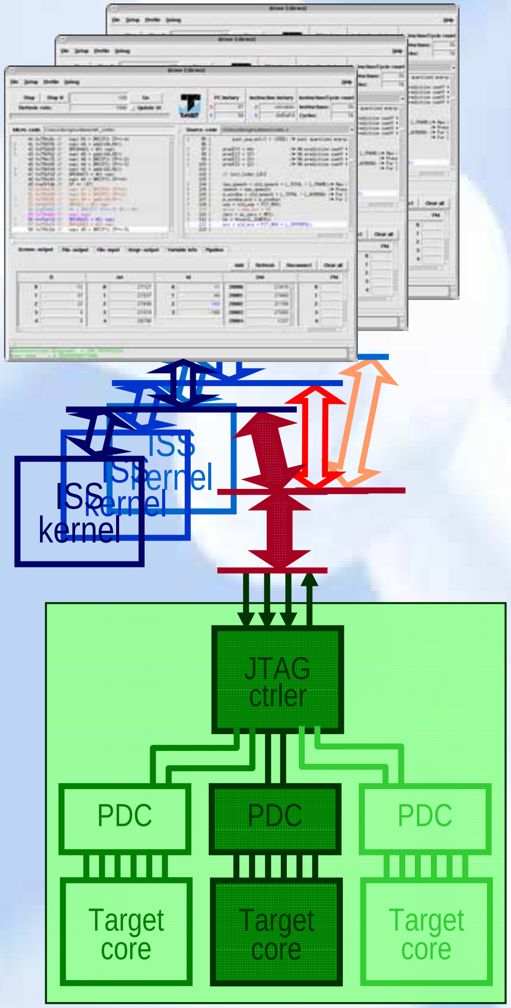
- **Single tool suite for all ASIPs in heterogeneous MPSoC**
- **System partitioning**
  - **Heterogeneous MPSoC:** often follows naturally from system-level block diagram



  - **Homogeneous MPSoC:** spatial & temporal mapping, taking into account NoC and RTOS → cf. Pier Paolucci's talk on Friday [Shapes project]
- **System specification**
  - **SystemC:** system architecture and communication
  - **C/C++:** functional blocks
- **Inter-processor communication**
  - C compiler supports memory-mapped I/O
  - Channels may have varying latency and wait states
  - HdS implements high-level communication protocols in C code
  - Memory and communication interface APIs in ASIP's ISS and HDL model

# Multi-processor aspects (2/2)

## Multi-processor simulation and debugging



### Instruction-set simulation

- ISS: graphical debugger and simulation kernel communicate via API

### On-chip debugging

- Generate Processor Debug Controller and JTAG I/F in HDL, next to target core
- Communication with graphical debugger via API, using TCP/IP sockets

### Multi-core simulation and debugging

- APIs for data communication
- Synchronisation of cores, e.g. local or global breakpoints

# Agenda

- SoC design trends
- Retargetable tool suite for ASIPs
- Multi-processor aspects
- Business model: IP or EDA?

# Business model : IP or EDA ?

| Traditional IP vendor’s model<br>**Per-chip royalty** | EDA vendor’s model<br>**License fee for retargetable tool suite** |
|---|---|
| More easily accepted for:<br>◣ Complete processor eco-systems: core + tools + application libraries + peripherals<br>◣ Lower-volume markets | More easily accepted for:<br>◣ Applications benefiting from proliferation of cores: MPSoC (Sea-of-Cores), fast design respins<br>◣ Higher-volume markets |
| Restricts differentiation between competing IP users, unless made configurable | Encourages differentiation between competing IP users |
| Configurable IP cores: how to defend royalties if user embeds proprietary application knowledge? | Supports creation of IP embedding proprietary application knowledge |

⬇

- Requires true retargetable tool suite
- Could have same impact as logic synthesis in nineties

# Conclusion

## Heterogeneous MPSoCs using ASIPs

- Important paradigm to implement future complex systems
- Retargetable tool suite is key to design such MPSoCs
  - Architectural exploration, hardware design & verification, software compilation & debugging
  - Novel processor modelling and compiler technology
- EDA business model fits well