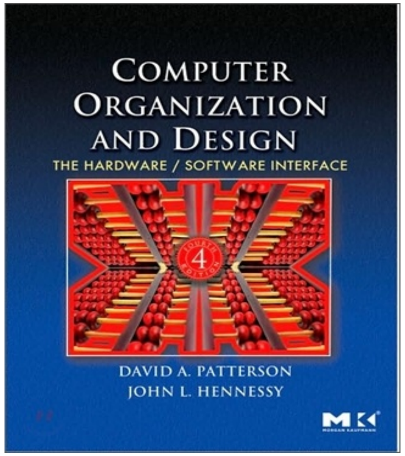
# Computer Architecture

## Lecture 10  Embedded Computer Architecture

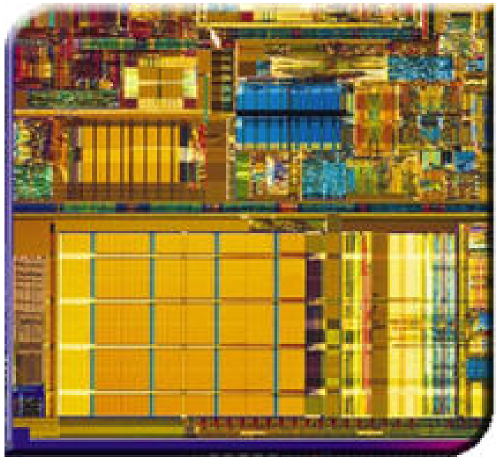


COMPUTER ORGANIZATION AND DESIGN

THE HARDWARE / SOFTWARE INTERFACE

DAVID A. PATTERSON

JOHN L. HENNESSY



**Prof. Jongmyon Kim**

울산대학교 UNIVERSITY OF ULSAN

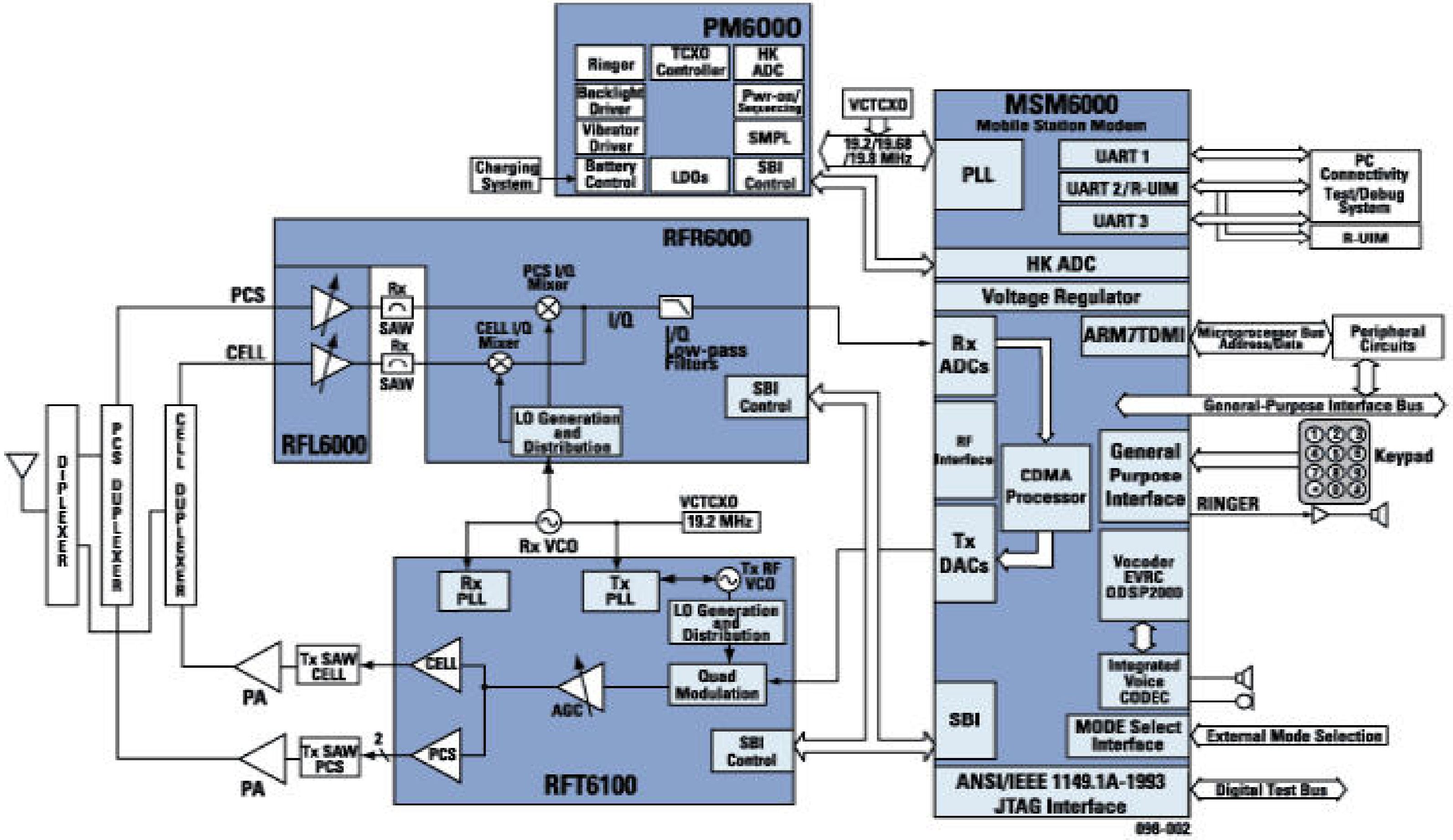
# Contents

- Introduction
- Examples of Embedded Computer Architecture

# What is an embedded processor architecture?

# Embedded processors in an embedded system

- 3G mobile communication (CDMA2000)

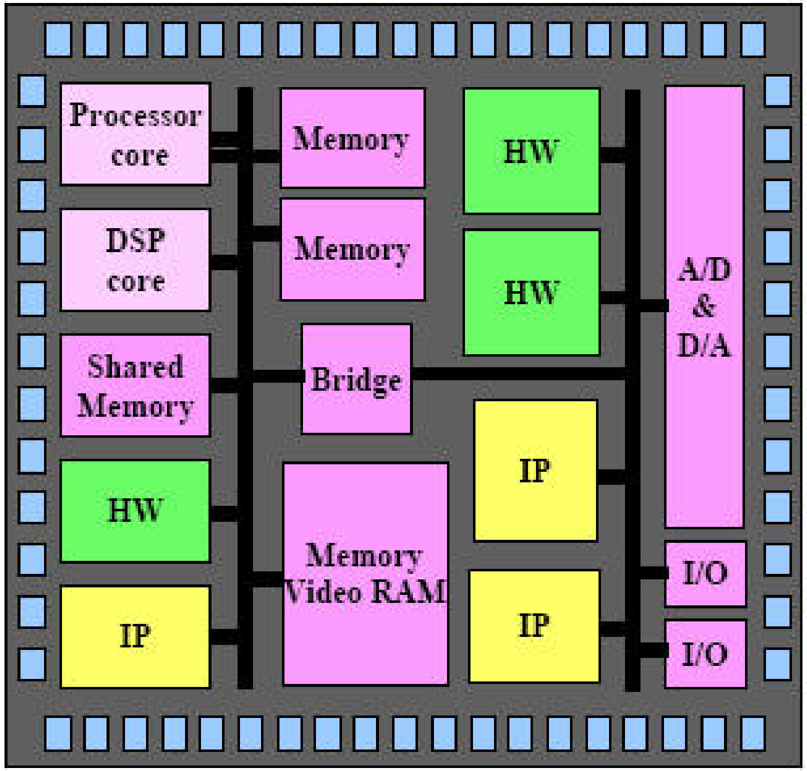


# Issues and trend

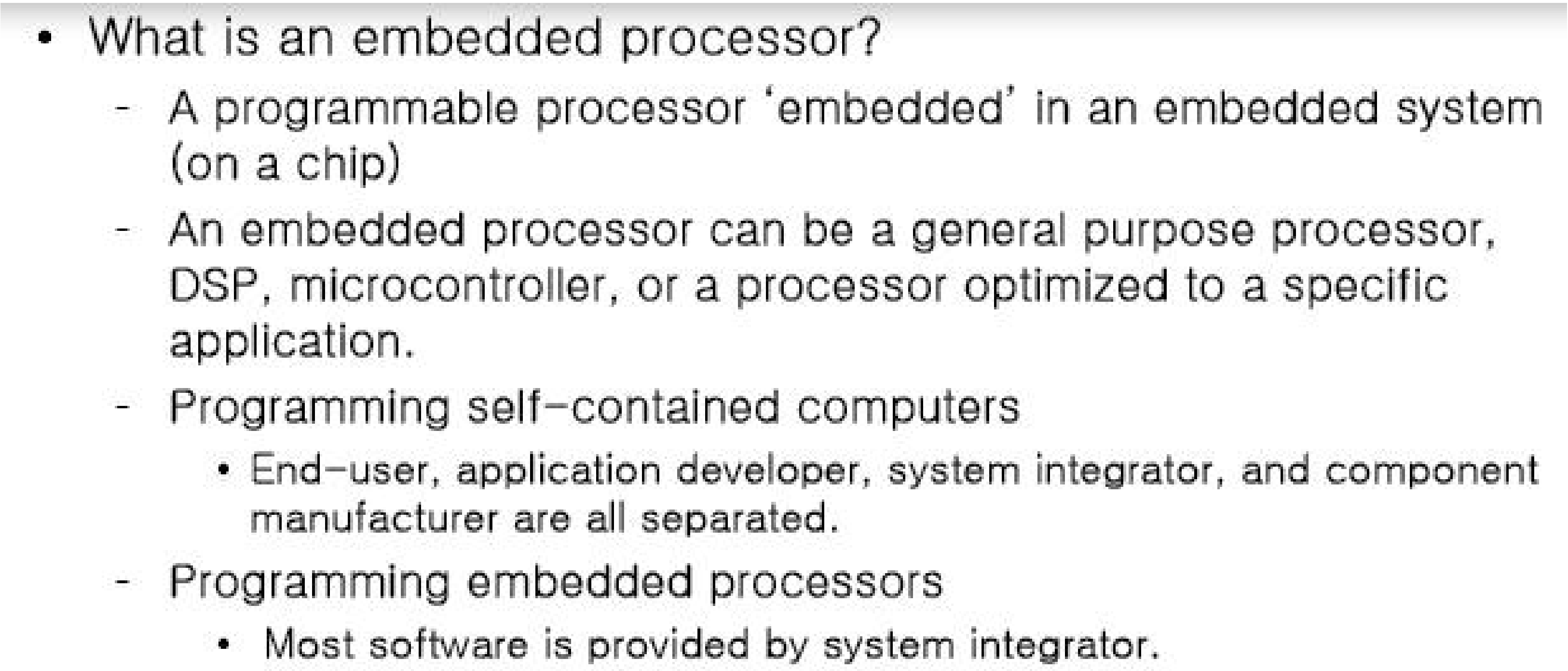
- Issues
  - Time-to-market
  - Complexity
  - Performance
  - Low power
  - Reliability
  - Size/weight
  - Cost
  - Etc.
- Paradigm shift toward software
  - Microprocessor performance rapidly increase
  - Flexibility, upgradability
  - Low cost

# Both HW and SW

- The system functionality is partitioned into HW, SW, and the Interface.
- SW part runs on programmable processors (embedded processors).
- HW part runs on IP blocks or custom designed HW blocks.



# Embedded processor

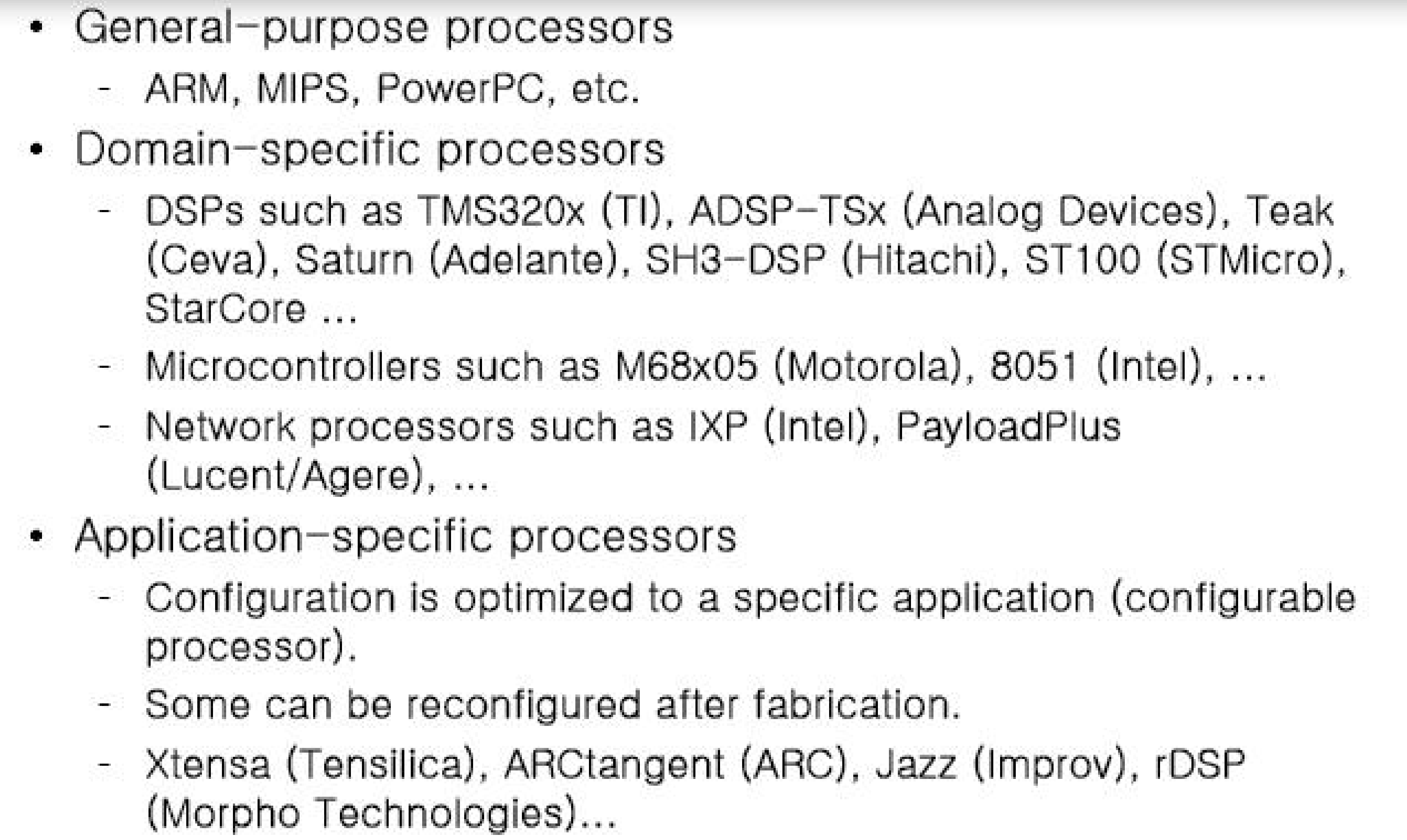


- What is an embedded processor?
  - A programmable processor 'embedded' in an embedded system (on a chip)
  - An embedded processor can be a general purpose processor, DSP, microcontroller, or a processor optimized to a specific application.
  - Programming self-contained computers
    - End-user, application developer, system integrator, and component manufacturer are all separated.
  - Programming embedded processors
    - Most software is provided by system integrator.

# Taxonomy of embedded processors

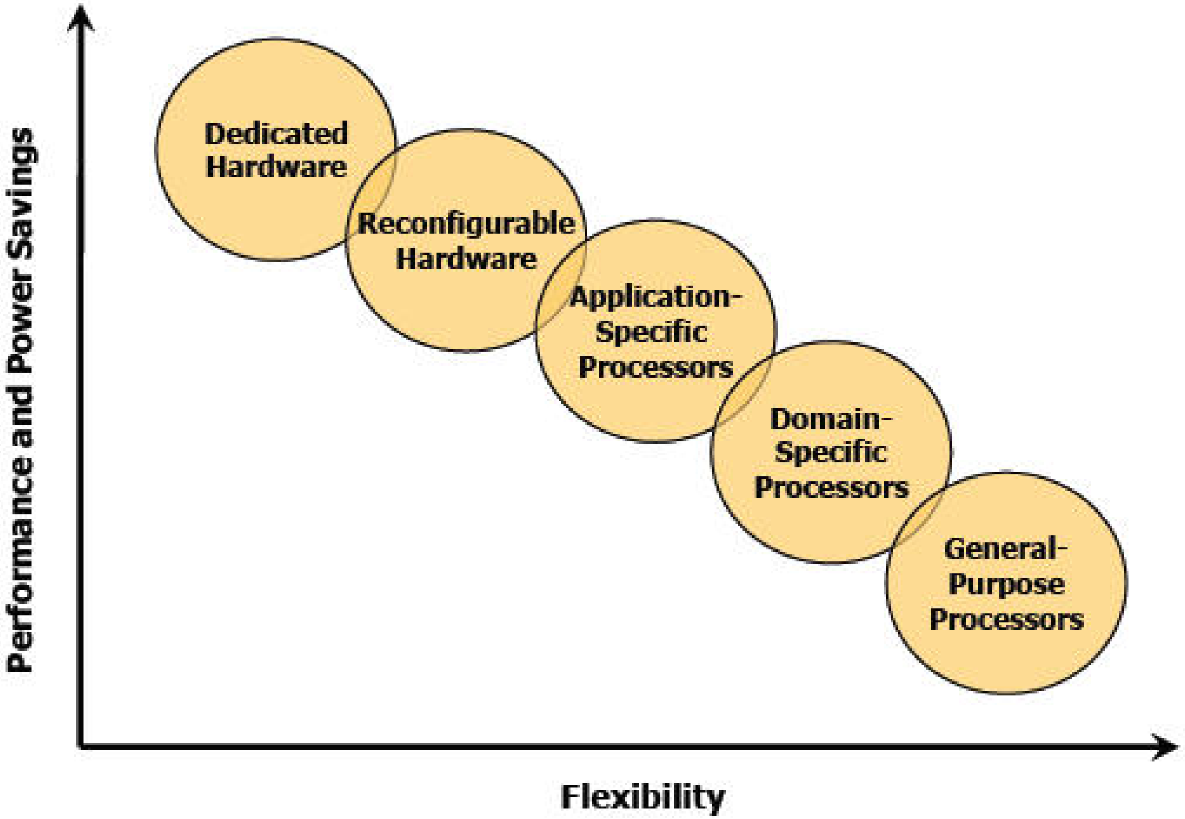
- Embedded processors are categorized according to the following aspects
  - Domain of target applications
  - Architecture
  - Configurability

# Domain of target applications

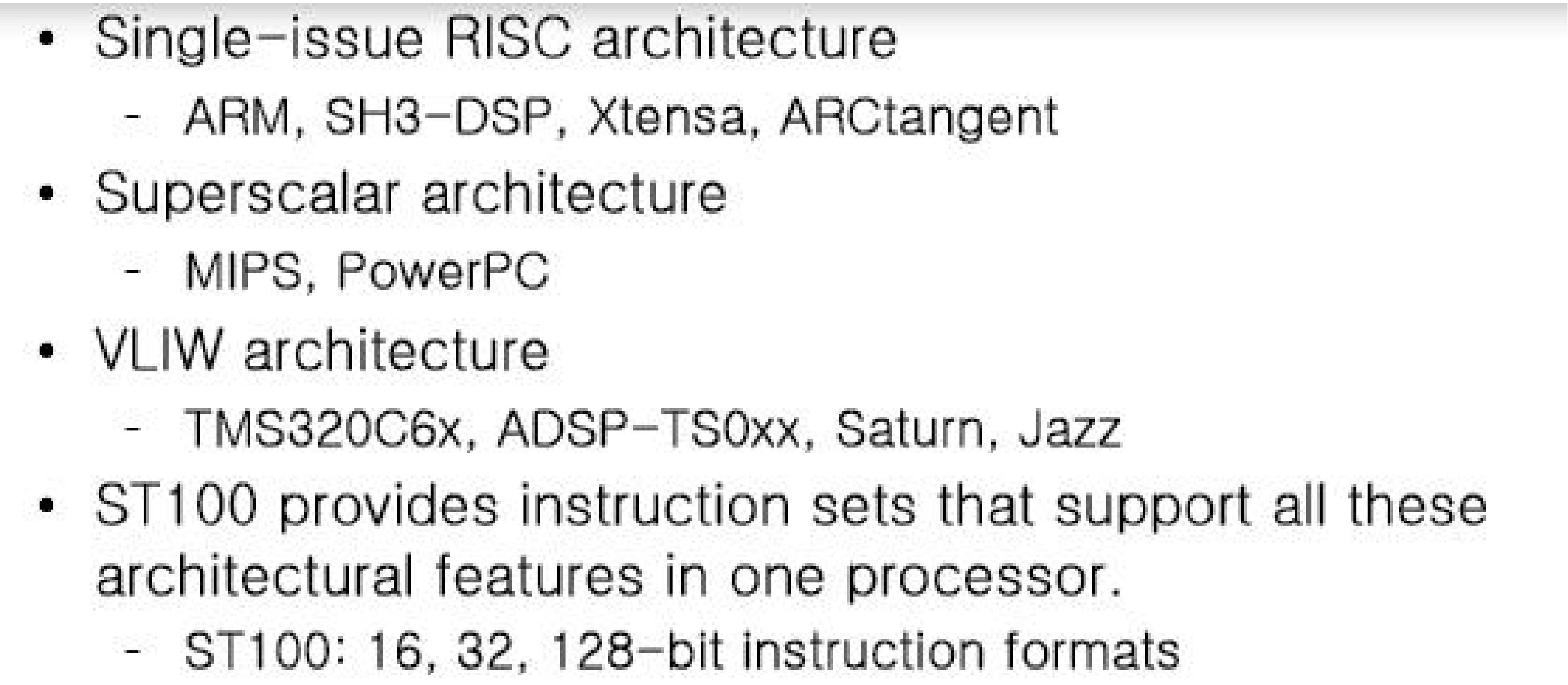


- General-purpose processors
  - ARM, MIPS, PowerPC, etc.
- Domain-specific processors
  - DSPs such as TMS320x (TI), ADSP-TSx (Analog Devices), Teak (Ceva), Saturn (Adelante), SH3-DSP (Hitachi), ST100 (STMicro), StarCore ...
  - Microcontrollers such as M68x05 (Motorola), 8051 (Intel), ...
  - Network processors such as IXP (Intel), PayloadPlus (Lucent/Agere), ...
- Application-specific processors
  - Configuration is optimized to a specific application (configurable processor).
  - Some can be reconfigured after fabrication.
  - Xtensa (Tensilica), ARCtangent (ARC), Jazz (Improv), rDSP (Morpho Technologies)...

# Tradeoffs due to architectural choice

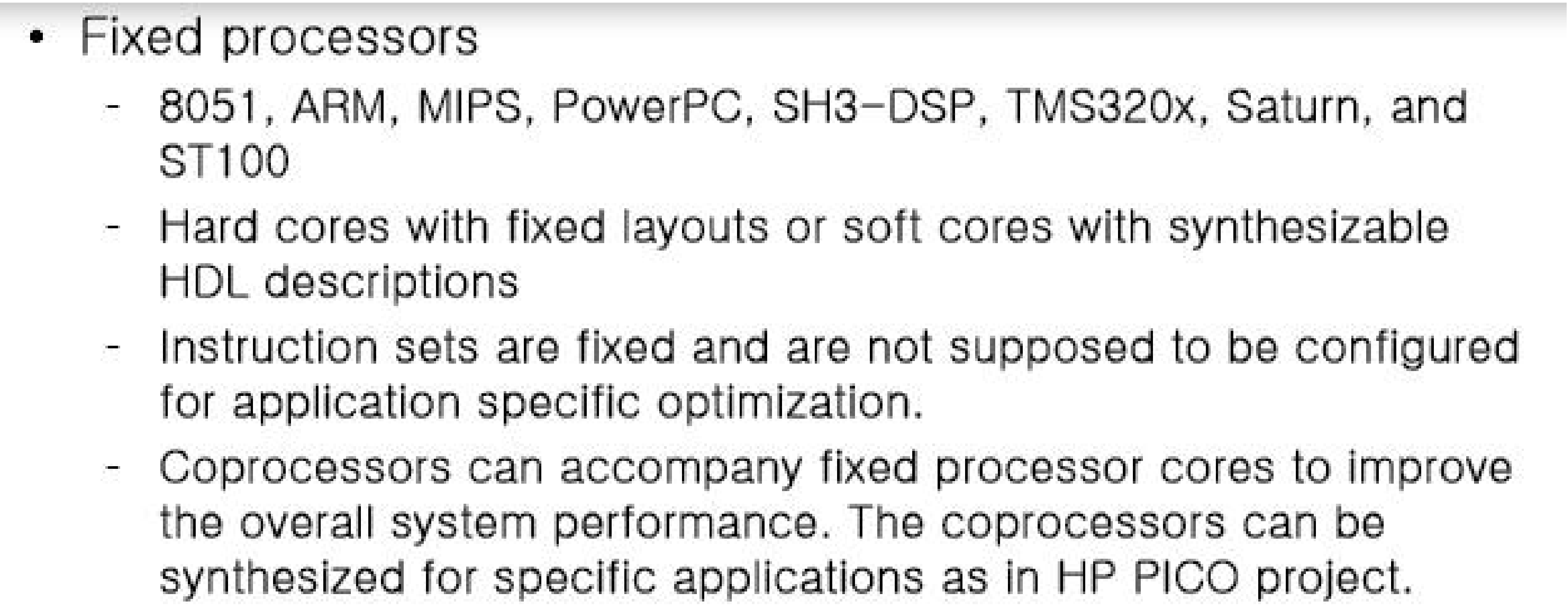

Dedicated Hardware
Reconfigurable Hardware
Application-Specific Processors
Domain-Specific Processors
General-Purpose Processors
Performance and Power Savings
Flexibility

# Architecture



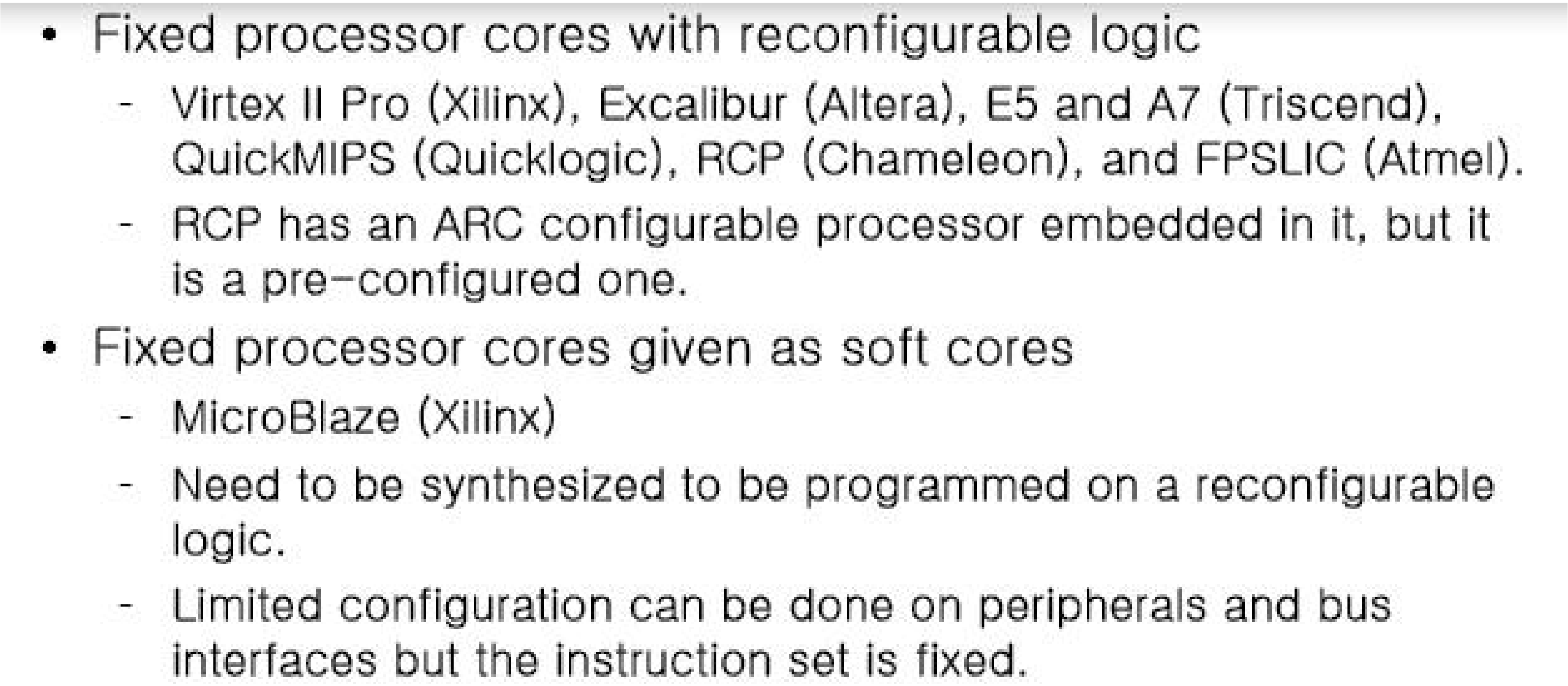
- Single-issue RISC architecture
  - ARM, SH3-DSP, Xtensa, ARCtangent
- Superscalar architecture
  - MIPS, PowerPC
- VLIW architecture
  - TMS320C6x, ADSP-TS0xx, Saturn, Jazz
- ST100 provides instruction sets that support all these architectural features in one processor.
  - ST100: 16, 32, 128-bit instruction formats

# Configurability



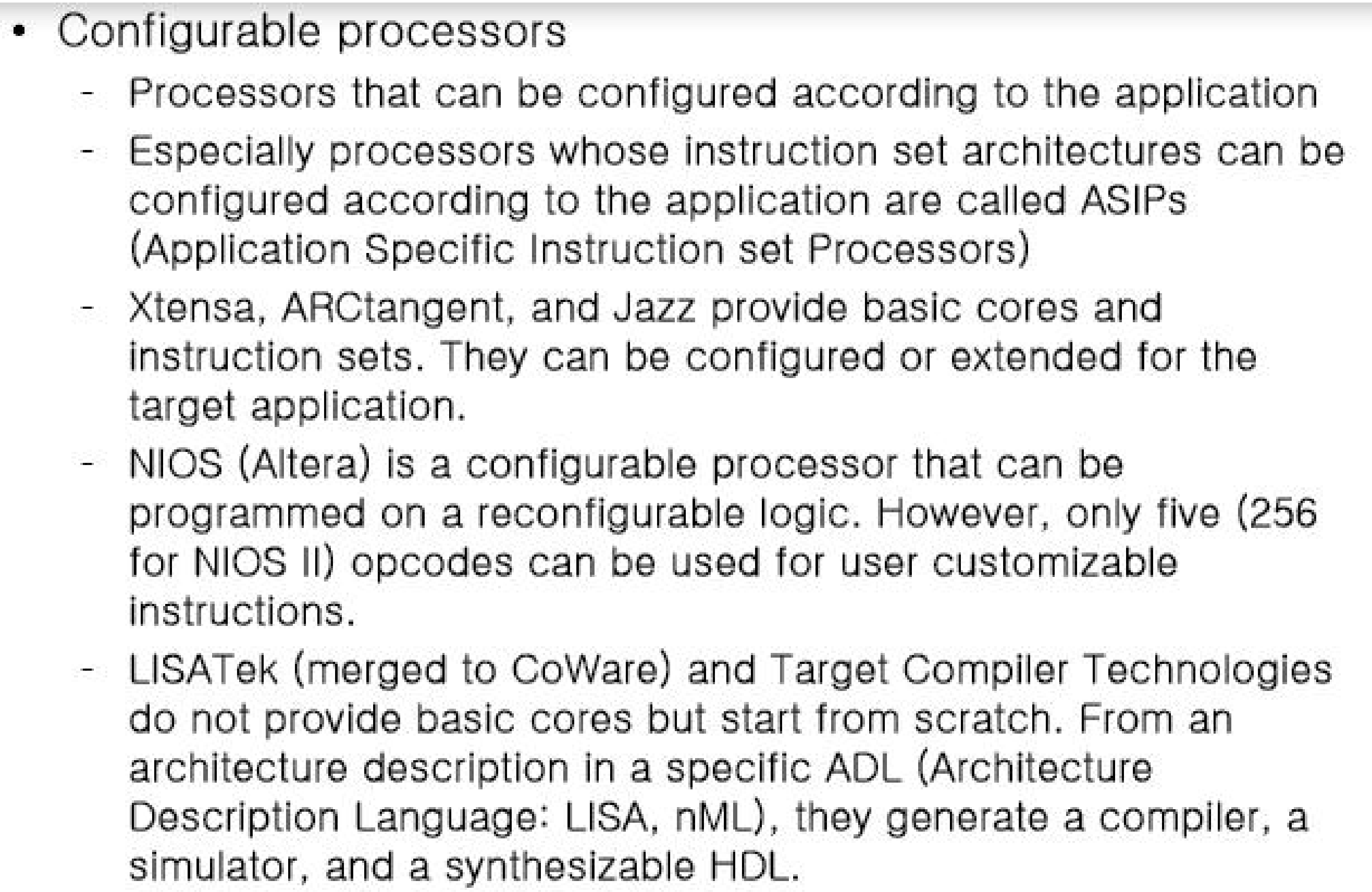
- Fixed processors
  - 8051, ARM, MIPS, PowerPC, SH3-DSP, TMS320x, Saturn, and ST100
  - Hard cores with fixed layouts or soft cores with synthesizable HDL descriptions
  - Instruction sets are fixed and are not supposed to be configured for application specific optimization.
  - Coprocessors can accompany fixed processor cores to improve the overall system performance. The coprocessors can be synthesized for specific applications as in HP PICO project.

# Configurability (cont’d)



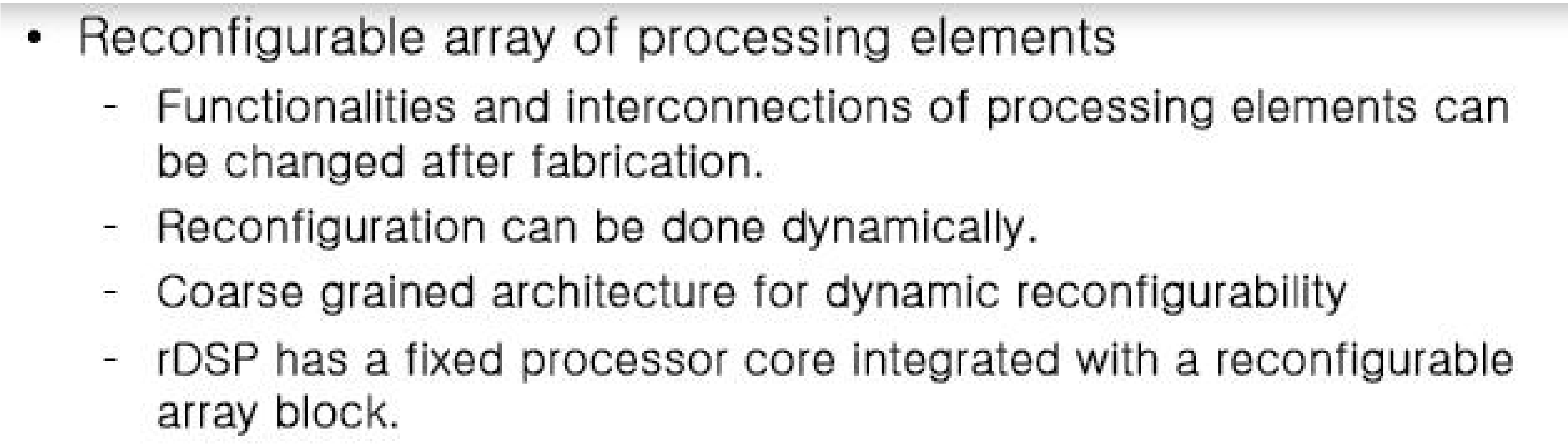
- Fixed processor cores with reconfigurable logic
  - Virtex II Pro (Xilinx), Excalibur (Altera), E5 and A7 (Triscend), QuickMIPS (Quicklogic), RCP (Chameleon), and FPSLIC (Atmel).
  - RCP has an ARC configurable processor embedded in it, but it is a pre-configured one.
- Fixed processor cores given as soft cores
  - MicroBlaze (Xilinx)
  - Need to be synthesized to be programmed on a reconfigurable logic.
  - Limited configuration can be done on peripherals and bus interfaces but the instruction set is fixed.

# Configurability (cont’d)



- Configurable processors
  - Processors that can be configured according to the application
  - Especially processors whose instruction set architectures can be configured according to the application are called ASIPs (Application Specific Instruction set Processors)
  - Xtensa, ARCtangent, and Jazz provide basic cores and instruction sets. They can be configured or extended for the target application.
  - NIOS (Altera) is a configurable processor that can be programmed on a reconfigurable logic. However, only five (256 for NIOS II) opcodes can be used for user customizable instructions.
  - LISATek (merged to CoWare) and Target Compiler Technologies do not provide basic cores but start from scratch. From an architecture description in a specific ADL (Architecture Description Language: LISA, nML), they generate a compiler, a simulator, and a synthesizable HDL.

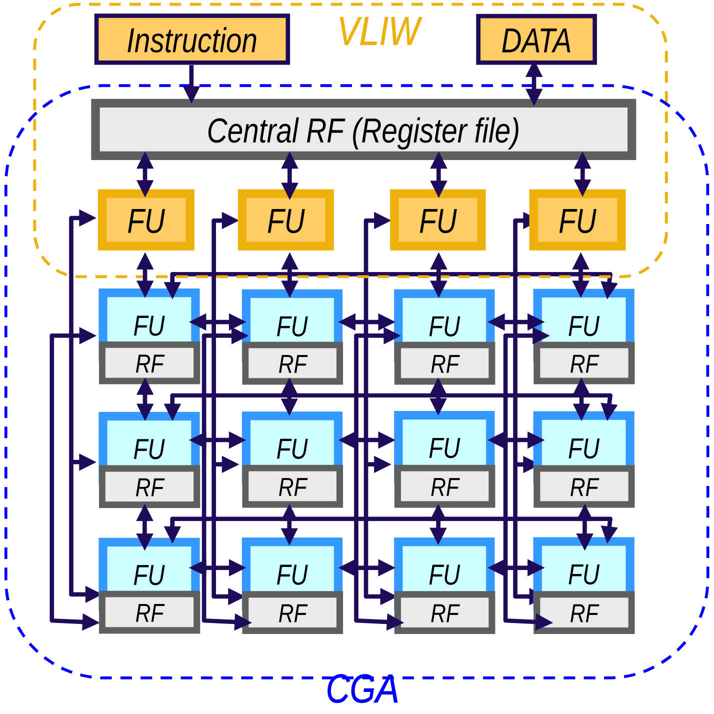
# Configurability (cont'd)

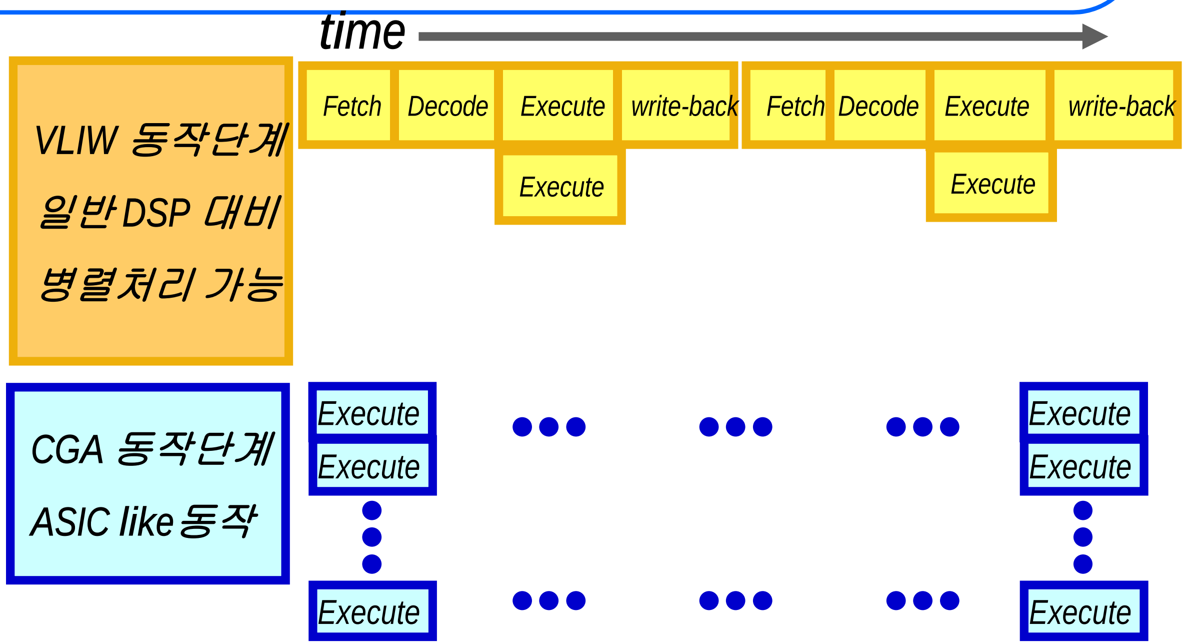


- Reconfigurable array of processing elements
  - Functionalities and interconnections of processing elements can be changed after fabrication.
  - Reconfiguration can be done dynamically.
  - Coarse grained architecture for dynamic reconfigurability
  - rDSP has a fixed processor core integrated with a reconfigurable array block.

# Reconfigurable processor

- ***RP (Reconfigurable Processor) 구조***
  - *DSP + co-processor 형태의 array 구조가 일반적*
- *Coarse-Grained RP*
  - *VLIW(DSP)와 CGA (loop 가속기)의 중첩형태*
  - *자원공유(RF, FU)에 의한 data transfer 최소화로 성능향상, 저전력, 소형화*
  - *VLIW 모드(control flow가속) + CGA모드(데이터 연산 가속) 역할 분담*
  - *CGA모드에서 SW pipeline 기법을 사용하여 가속능력 극대화*
  - *임의의 HW구조설계 가능 및 compiler 수정 없이 적용가능*

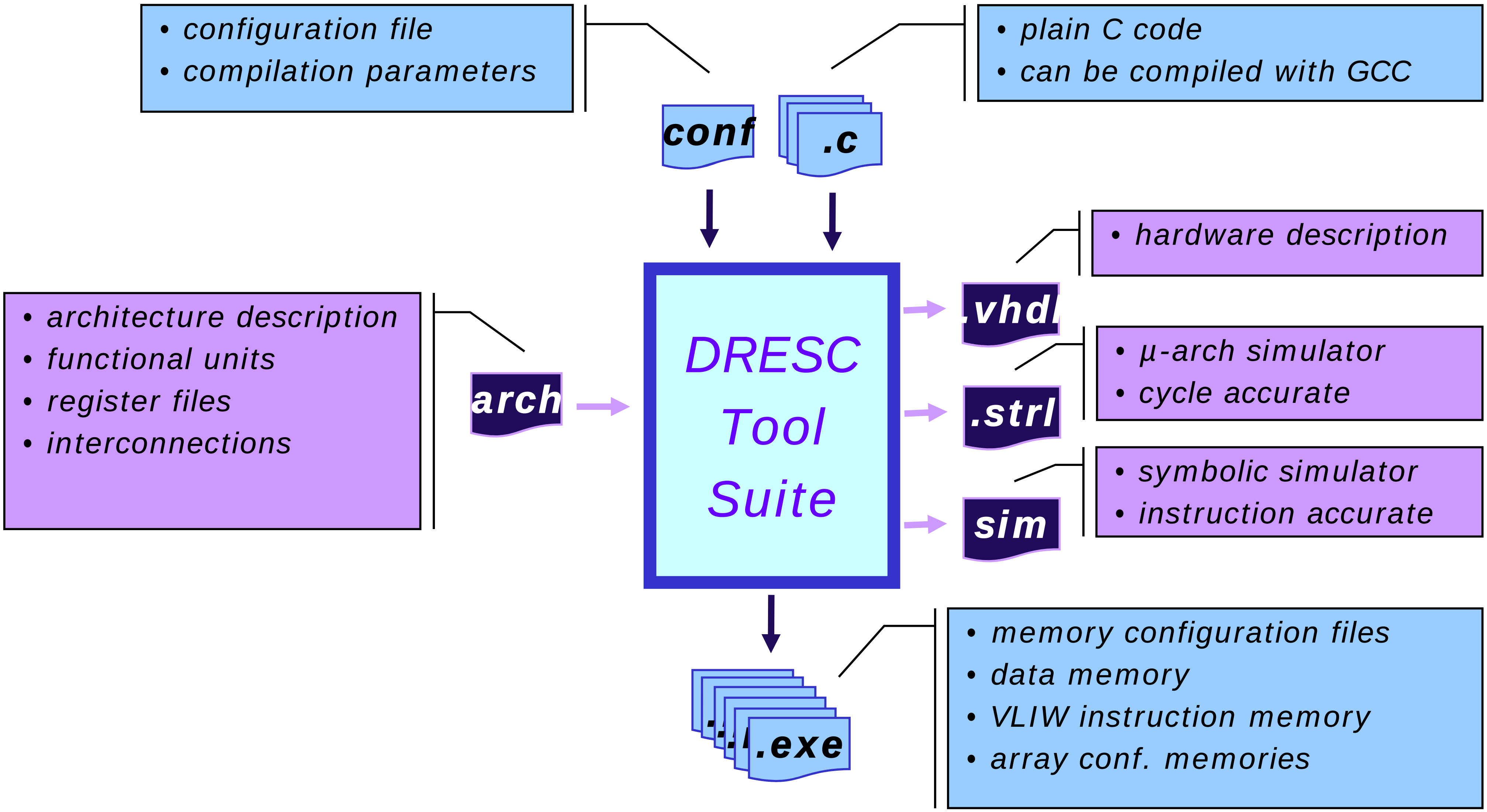




*VLIW: Very Long Instruction Word   CGA: Coarse Grain Array*

*RF: Register File  FU: Functional Unit*

# Reconfigurable processor tool suite



- configuration file
- compilation parameters

- plain C code
- can be compiled with GCC

- architecture description
- functional units
- register files
- interconnections

- hardware description

- µ-arch simulator
- cycle accurate

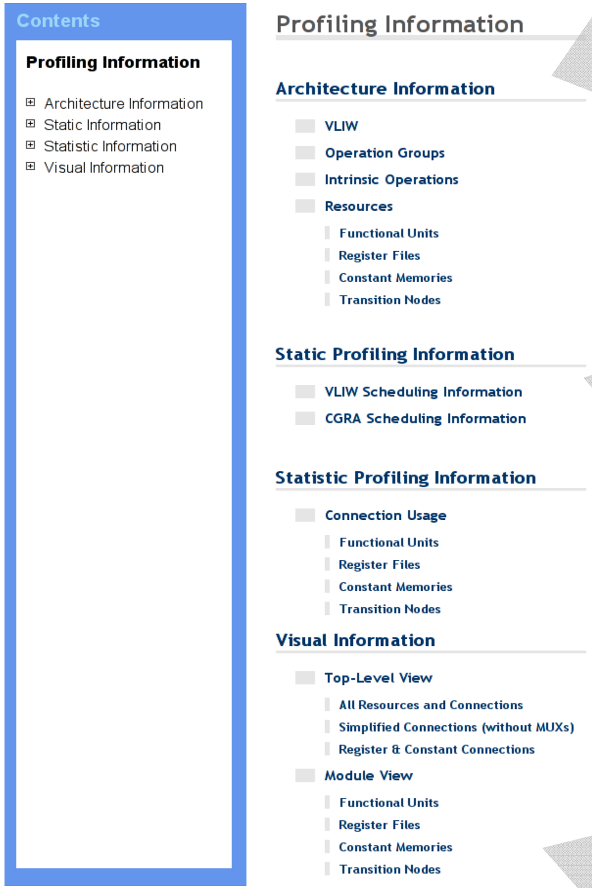
- symbolic simulator
- instruction accurate

- memory configuration files
- data memory
- VLIW instruction memory
- array conf. memories

*DRESC: Dynamically Reconfigurable Embedded System Compiler*

# Architectural Exploration with an Profiler

## Profiling Tool



Contents

**Profiling Information**

- ⊞ Architecture Information
- ⊞ Static Information
- ⊞ Statistic Information
- ⊞ Visual Information

Profiling Information

Architecture Information

- VLIW
- Operation Groups
- Intrinsic Operations
- Resources
  - Functional Units
  - Register Files
  - Constant Memories
  - Transition Nodes

Static Profiling Information

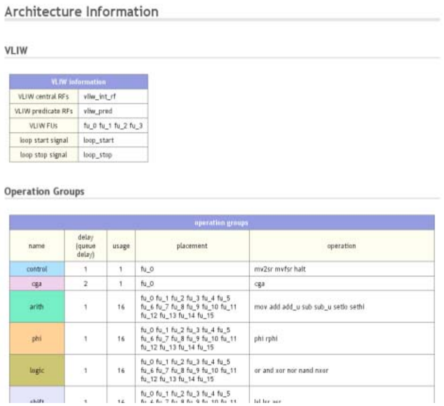
- VLIW Scheduling Information
- CGRA Scheduling Information

Statistic Profiling Information

- Connection Usage
  - Functional Units
  - Register Files
  - Constant Memories
  - Transition Nodes

Visual Information

- Top-Level View
  - All Resources and Connections
  - Simplified Connections (without MUXs)
  - Register & Constant Connections
- Module View
  - Functional Units
  - Register Files
  - Constant Memories
  - Transition Nodes



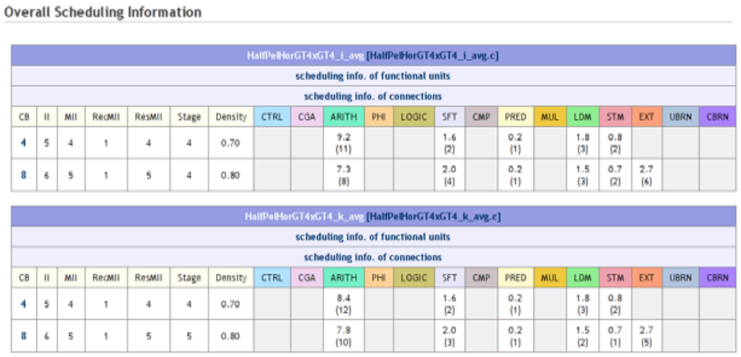
Architecture Information

VLIW

| VLIW information | |
|---|---|
| VLIW central RFs | vliw_int_rf |
| VLIW predicate RFs | vliw_pred |
| VLIW FUs | fu_0 fu_1 fu_2 fu_3 |
| loop start signal | loop_start |
| loop stop signal | loop_stop |

Operation Groups

| operation groups | | | | |
|---|---|---|---|---|
| name | delay (queue delay) | usage | placement | operation |
| control | 1 | 1 | fu_0 | mv2sr mvfsr halt |
| cga | 2 | 1 | fu_0 | cga |
| arith | 1 | 16 | fu_0 fu_1 fu_2 fu_3 fu_4 fu_5 fu_6 fu_7 fu_8 fu_9 fu_10 fu_11 fu_12 fu_13 fu_14 fu_15 | mov add add_u sub sub_u setlo sethi |
| phi | 1 | 16 | fu_0 fu_1 fu_2 fu_3 fu_4 fu_5 fu_6 fu_7 fu_8 fu_9 fu_10 fu_11 fu_12 fu_13 fu_14 fu_15 | phi rphi |
| logic | 1 | 16 | fu_0 fu_1 fu_2 fu_3 fu_4 fu_5 fu_6 fu_7 fu_8 fu_9 fu_10 fu_11 fu_12 fu_13 fu_14 fu_15 | or and xor nor nand nxor |
| shift | 1 | 16 | fu_0 fu_1 fu_2 fu_3 fu_4 fu_5 | lsl lsr asr |



Overall Scheduling Information

HalfPelHorGT4xGT4_i_avg [HalfPelHorGT4xGT4_i_avg.c]

scheduling info. of functional units

scheduling info. of connections

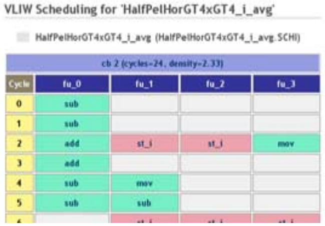
| CB | II | MII | RecMII | ResMII | Stage | Density | CTRL | CGA | ARITH | PHI | LOGIC | SFT | CMP | PRED | MUL | LDM | STM | EXT | UBRN | CBRN |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4 | 5 | 4 | 1 | 4 | 4 | 0.70 | | | 9.2 (11) | | | 1.6 (2) | | 0.2 (1) | | 1.8 (3) | 0.8 (2) | | | |
| 8 | 6 | 5 | 1 | 5 | 4 | 0.80 | | | 7.3 (8) | | | 2.0 (4) | | 0.2 (1) | | 1.5 (3) | 0.7 (2) | 2.7 (6) | | |

HalfPelHorGT4xGT4_k_avg [HalfPelHorGT4xGT4_k_avg.c]

scheduling info. of functional units

scheduling info. of connections

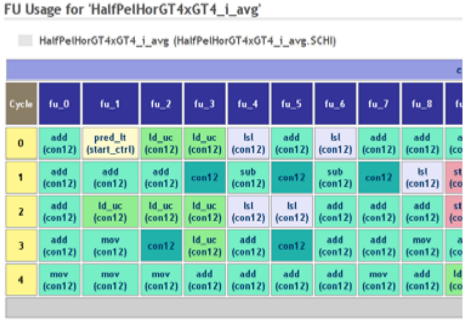
| CB | II | MII | RecMII | ResMII | Stage | Density | CTRL | CGA | ARITH | PHI | LOGIC | SFT | CMP | PRED | MUL | LDM | STM | EXT | UBRN | CBRN |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4 | 5 | 4 | 1 | 4 | 4 | 0.70 | | | 8.4 (12) | | | 1.6 (2) | | 0.2 (1) | | 1.8 (3) | 0.8 (2) | | | |
| 8 | 6 | 5 | 1 | 5 | 5 | 0.80 | | | 7.8 (10) | | | 2.0 (3) | | 0.2 (1) | | 1.5 (2) | 0.7 (1) | 2.7 (5) | | |



VLIW Scheduling for 'HalfPelHorGT4xGT4_i_avg'

HalfPelHorGT4xGT4_i_avg (HalfPelHorGT4xGT4_i_avg.SCHI)

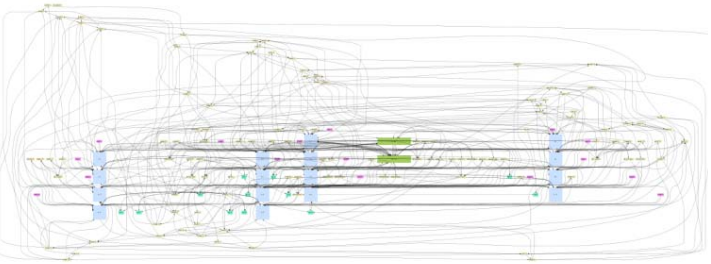
| cb 2 (cycles=24, density=2.33) | | | | |
|---|---|---|---|---|
| Cycle | fu_0 | fu_1 | fu_2 | fu_3 |
| 0 | sub | | | |
| 1 | sub | | | |
| 2 | add | st_i | st_i | mov |
| 3 | add | | | |
| 4 | sub | mov | | |
| 5 | sub | sub | | |

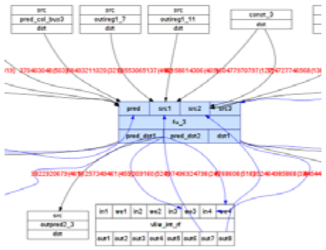


FU Usage for 'HalfPelHorGT4xGT4_i_avg'

HalfPelHorGT4xGT4_i_avg (HalfPelHorGT4xGT4_i_avg.SCHI)

| Cycle | fu_0 | fu_1 | fu_2 | fu_3 | fu_4 | fu_5 | fu_6 | fu_7 | fu_8 | fu |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | add (con12) | pred_lt (start_ctrl) | ld_uc (con12) | ld_uc (con12) | lsl (con12) | add (con12) | lsl (con12) | add (con12) | add (con12) | a (co |
| 1 | add (con12) | add (con12) | add (con12) | con12 | sub (con12) | con12 | sub (con12) | con12 | lsl (con12) | st (co |
| 2 | add (con12) | ld_uc (con12) | ld_uc (con12) | ld_uc (con12) | lsl (con12) | lsl (con12) | add (con12) | add (con12) | add (con12) | st (co |
| 3 | add (con12) | mov (con12) | con12 | ld_uc (con12) | add (con12) | con12 | add (con12) | add (con12) | mov (con12) | a (co |
| 4 | mov (con12) | mov (con12) | mov (con12) | add (con12) | add (con12) | add (con12) | add (con12) | mov (con12) | add (con12) | ld (co |



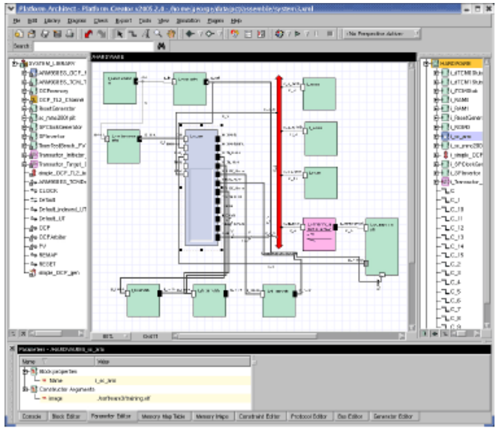


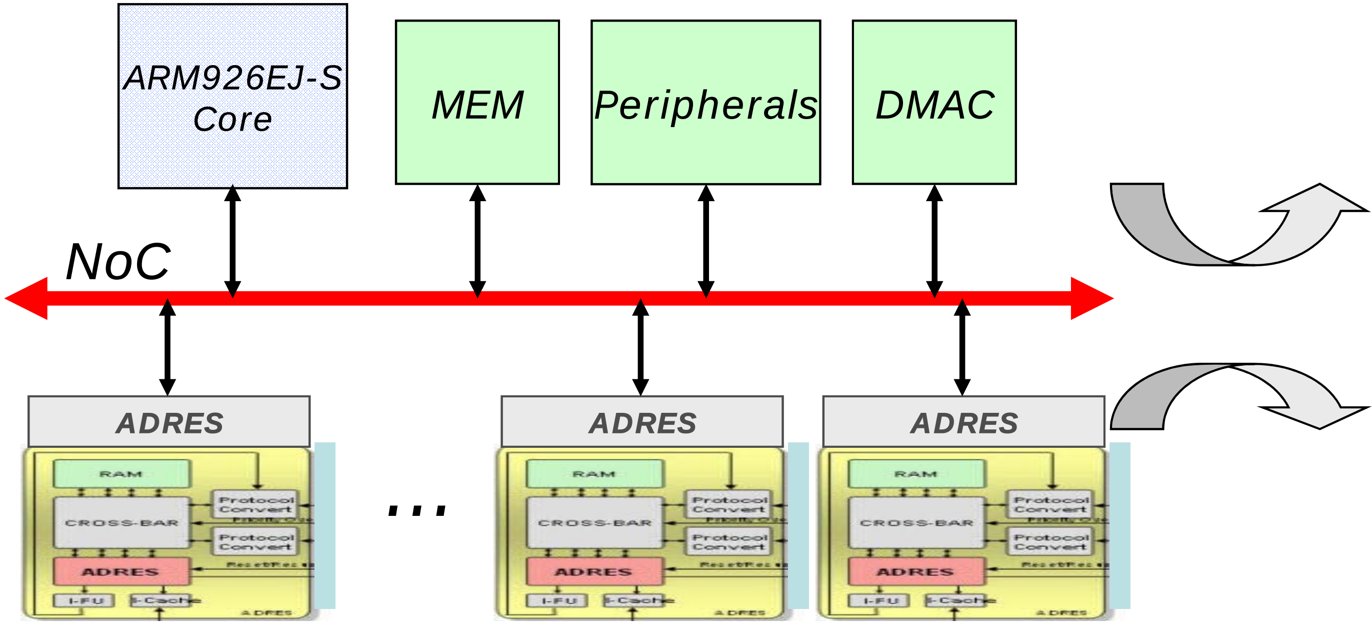
# Virtual Platform Design for MPSoC

## *Virtual Platform*

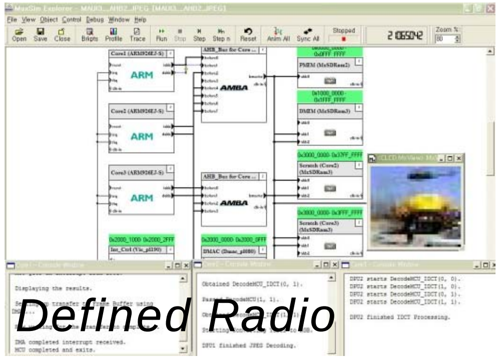
- *Verify developed IP's functionality on an existing virtual system*
- *Meet time-to-market by early verifying the target system on the higher level simulation*

CoWare The ESL Design Leader **ConversenSC**





ARM **SoC Designer**



*ADRES: Architecture for Dynamically Reconfigurable Embedded Systems*

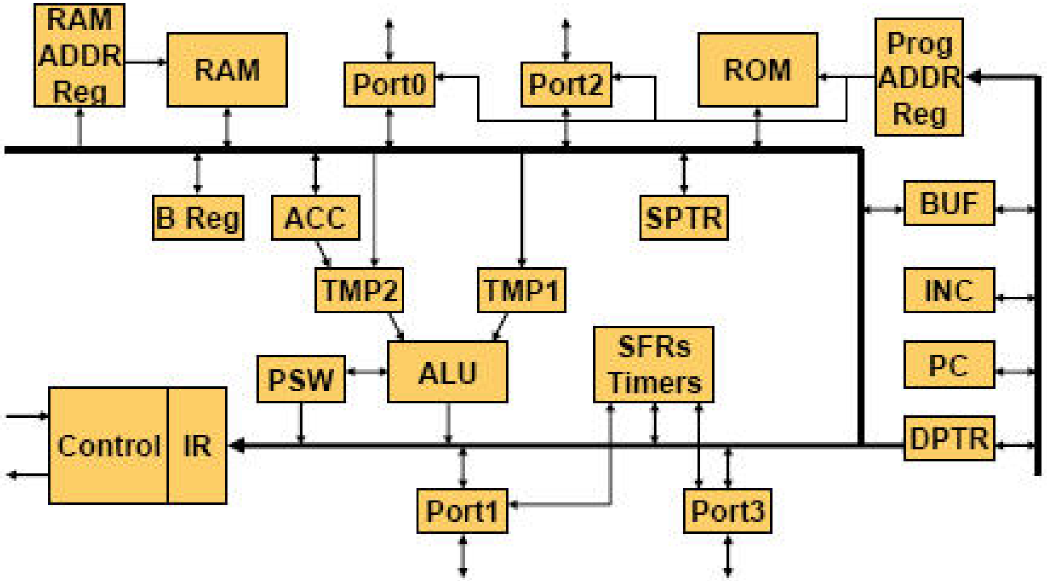
***Target Applications***: *Audio, Video, 3D Graphics, Software Defined Radio*

# Contents

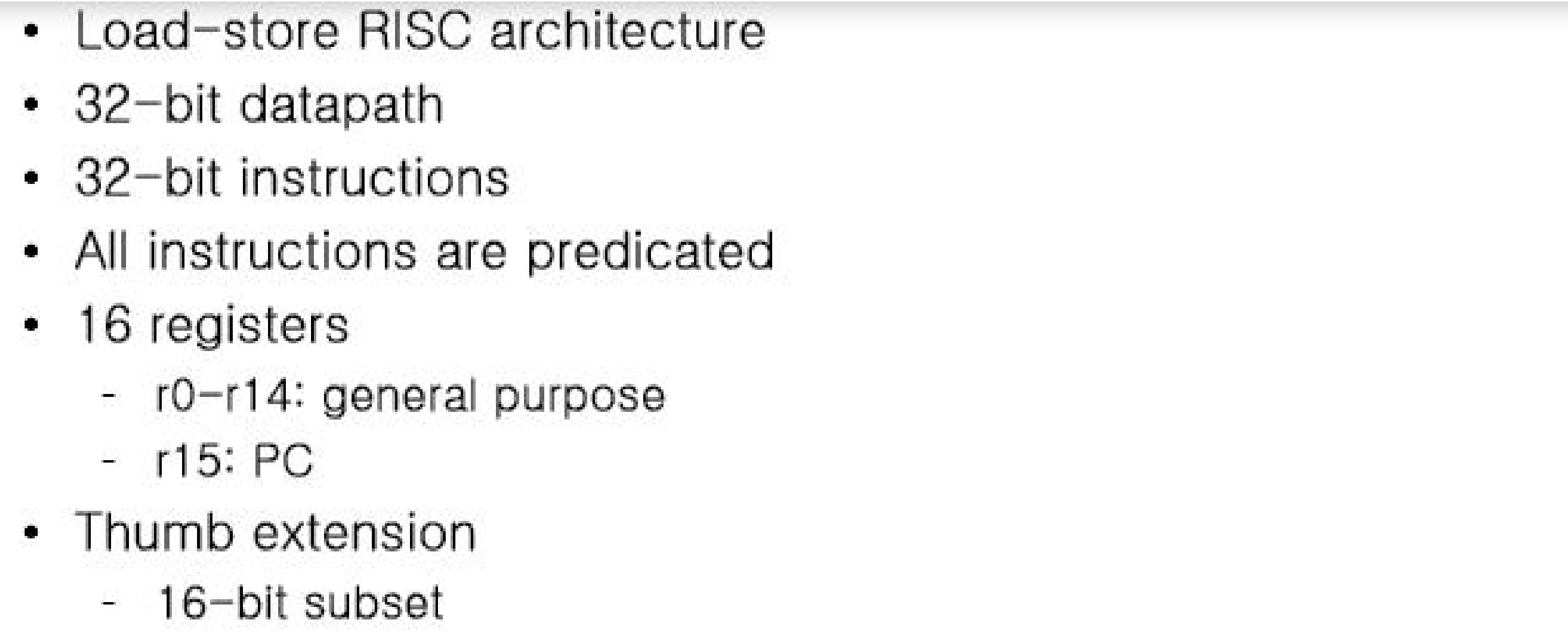
- Introduction
- Examples of Embedded Processors

# 8051

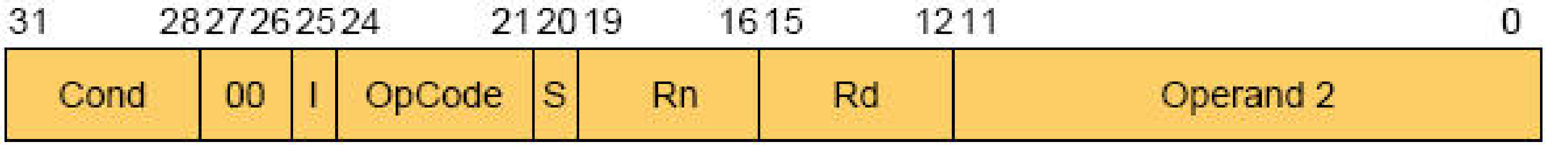
- Originally designed in the 1980’s by Intel
- One of the most popular embedded microcontrollers
- 8-bit CPU, on-chip memory (RAM, ROM), two 16-bit timer/counters, and four 8-bit I/O ports
- There are many variations



# ARM

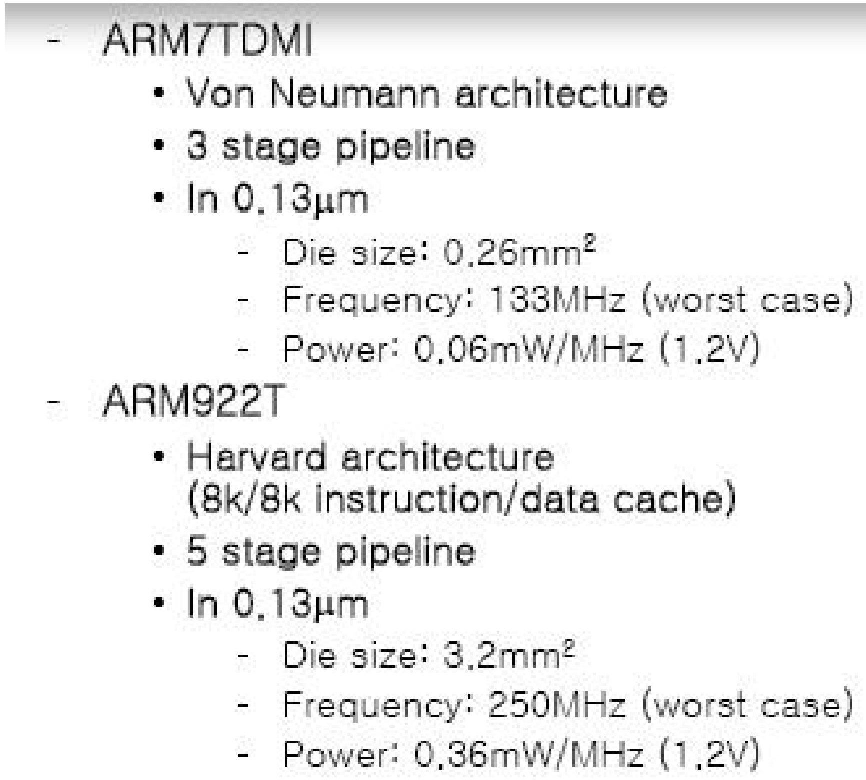


- Load-store RISC architecture
- 32-bit datapath
- 32-bit instructions
- All instructions are predicated
- 16 registers
  - r0-r14: general purpose
  - r15: PC
- Thumb extension
  - 16-bit subset

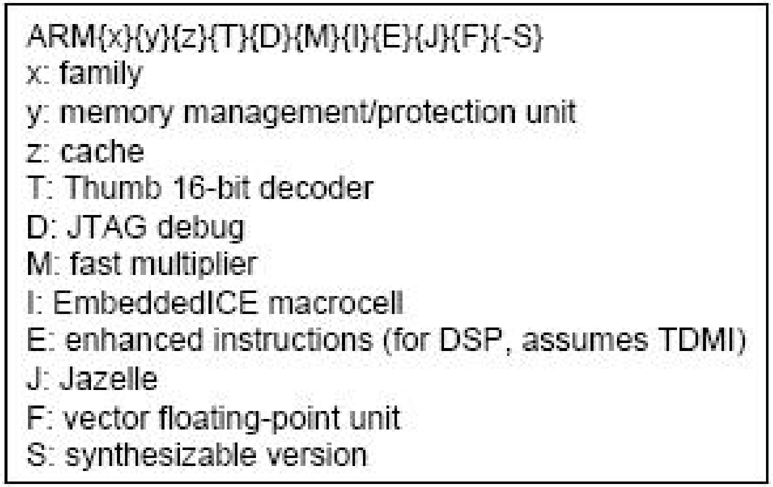


| 31 28 | 27 26 | 25 | 24 21 | 20 | 19 16 | 15 12 | 11 0 |
|---|---|---|---|---|---|---|---|
| Cond | 00 | I | OpCode | S | Rn | Rd | Operand 2 |

# ARM (cont'd)

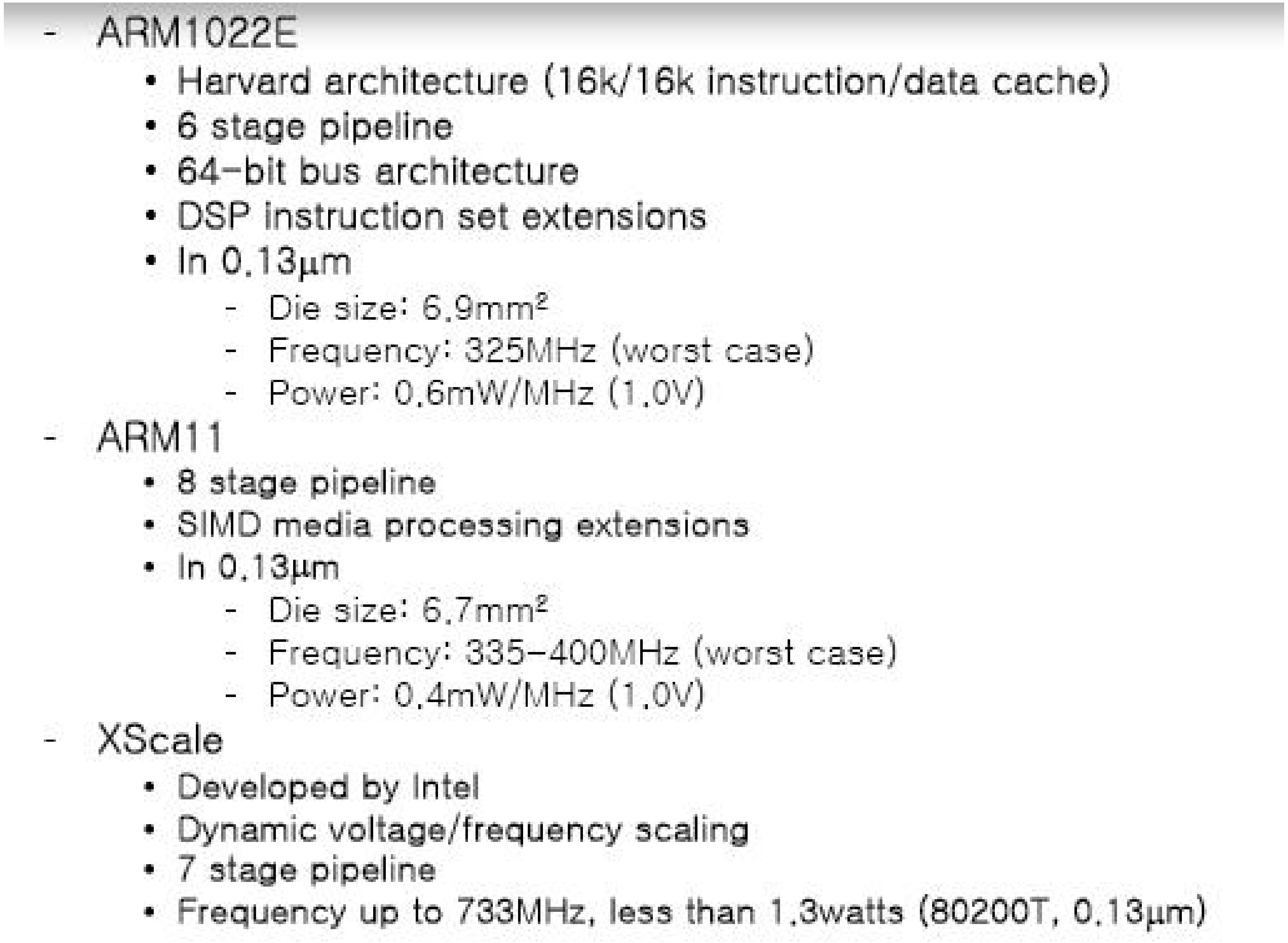


- ARM7TDMI
  - Von Neumann architecture
  - 3 stage pipeline
  - In 0.13$\mu$m
    - Die size: 0.26mm$^2$
    - Frequency: 133MHz (worst case)
    - Power: 0.06mW/MHz (1.2V)
- ARM922T
  - Harvard architecture
    (8k/8k instruction/data cache)
  - 5 stage pipeline
  - In 0.13$\mu$m
    - Die size: 3.2mm$^2$
    - Frequency: 250MHz (worst case)
    - Power: 0.36mW/MHz (1.2V)



ARM{x}{y}{z}{T}{D}{M}{I}{E}{J}{F}{-S}
x: family
y: memory management/protection unit
z: cache
T: Thumb 16-bit decoder
D: JTAG debug
M: fast multiplier
I: EmbeddedICE macrocell
E: enhanced instructions (for DSP, assumes TDMI)
J: Jazelle
F: vector floating-point unit
S: synthesizable version

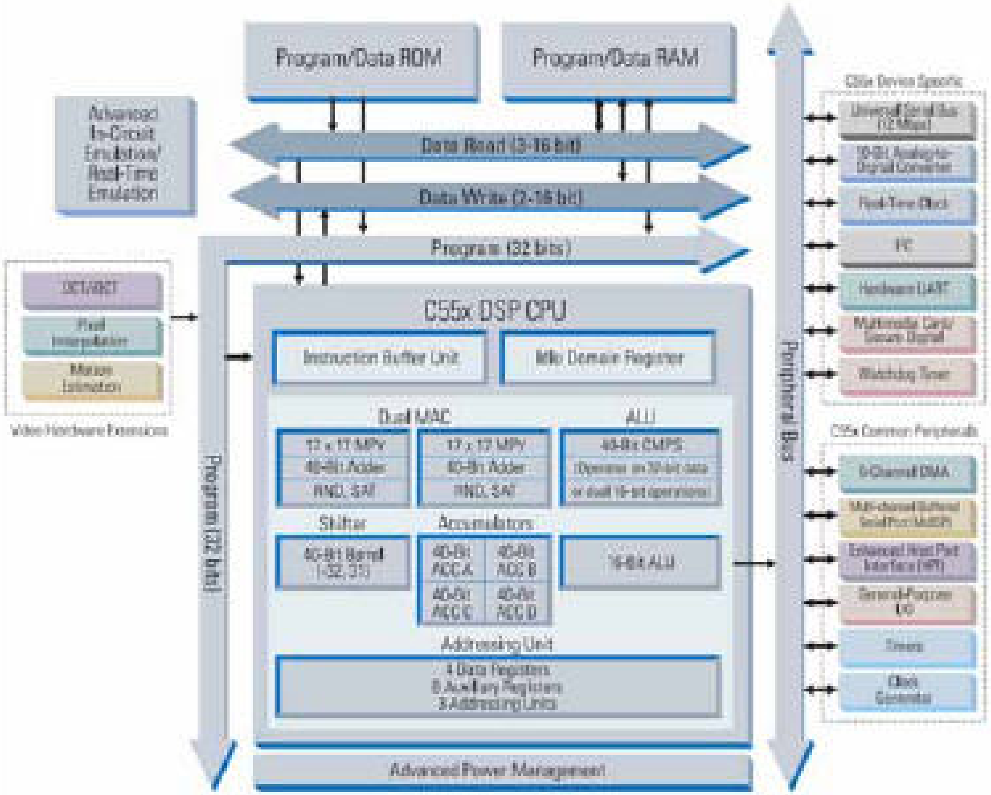
# ARM (cont’d)



- ARM1022E
  - Harvard architecture (16k/16k instruction/data cache)
  - 6 stage pipeline
  - 64-bit bus architecture
  - DSP instruction set extensions
  - In 0.13µm
    - Die size: 6.9mm$^2$
    - Frequency: 325MHz (worst case)
    - Power: 0.6mW/MHz (1.0V)
- ARM11
  - 8 stage pipeline
  - SIMD media processing extensions
  - In 0.13µm
    - Die size: 6.7mm$^2$
    - Frequency: 335-400MHz (worst case)
    - Power: 0.4mW/MHz (1.0V)
- XScale
  - Developed by Intel
  - Dynamic voltage/frequency scaling
  - 7 stage pipeline
  - Frequency up to 733MHz, less than 1.3watts (80200T, 0.13µm)

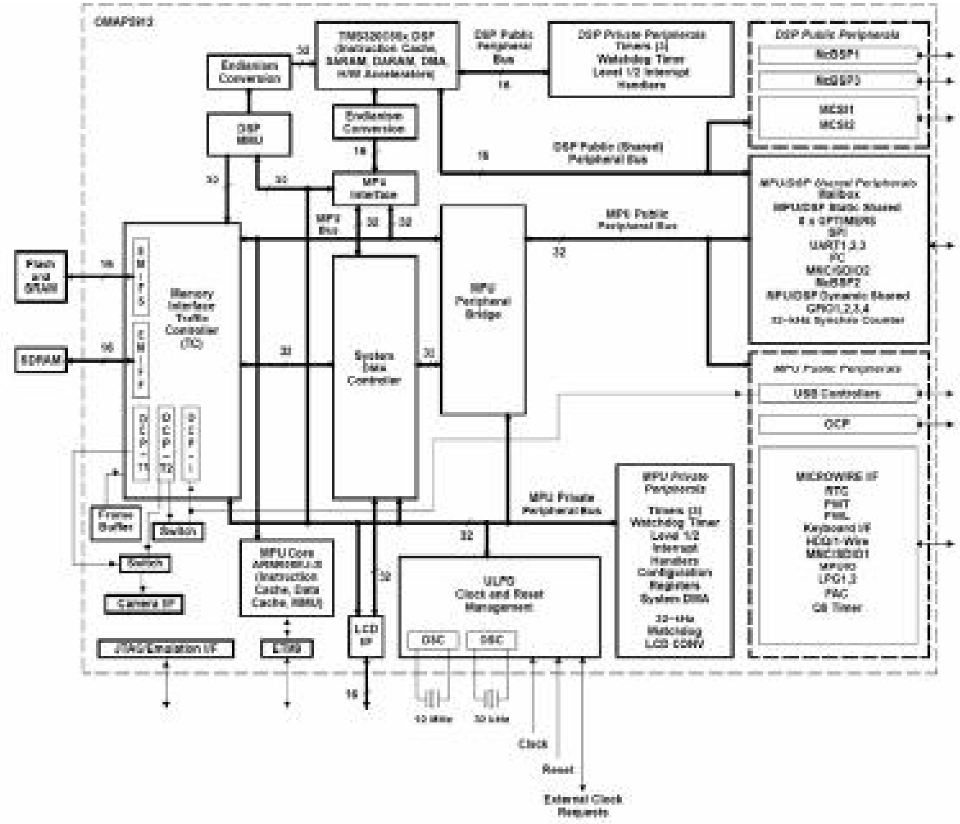
# TMS320C55x

- TMS320C55x
  - Fixed-point DSP core
  - Two multiply-accumulate (MAC) units, each capable of 17-bit x 17-bit multiplication in a single cycle
  - A central 40-bit arithmetic/logic unit (ALU) is supported by an additional 16-bit ALU.



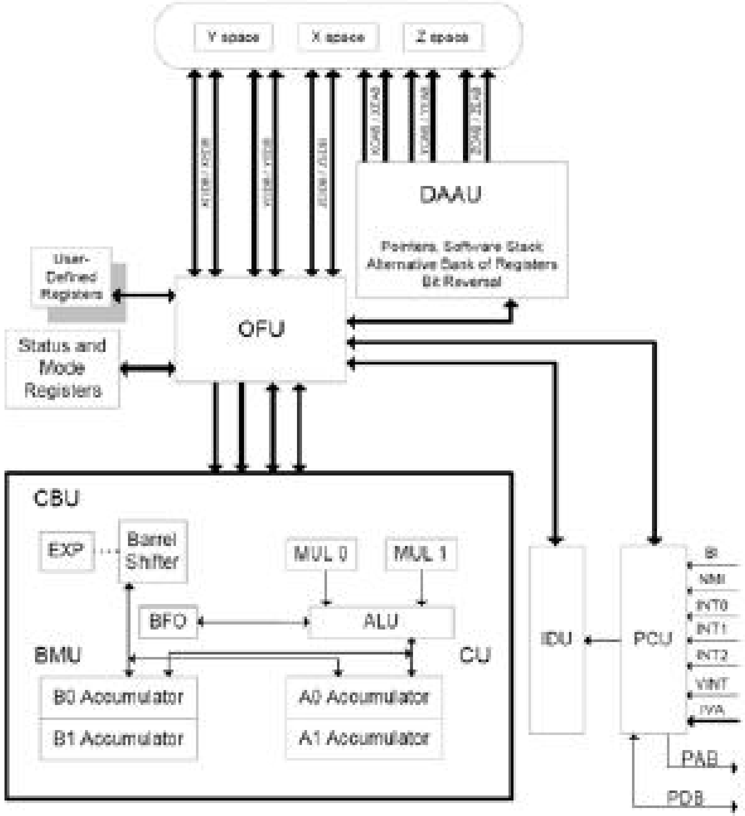
# OMAP5910 DSP Platform

- OMAP5912 DSP Platform
  - C55x DSP + ARM9 MPU
  - 0.13μm, 1.6V core power supply
  - 250KB shared internal SRAM
  - External memory interface
  - DMA controller
  - DSP peripherals
    - timers, serial ports, …
  - MPU peripherals
    - timers, USB, camera i/f keyboard i/f, LCD controller, …
  - Shared peripherals
    - UARTs, mailbox, …



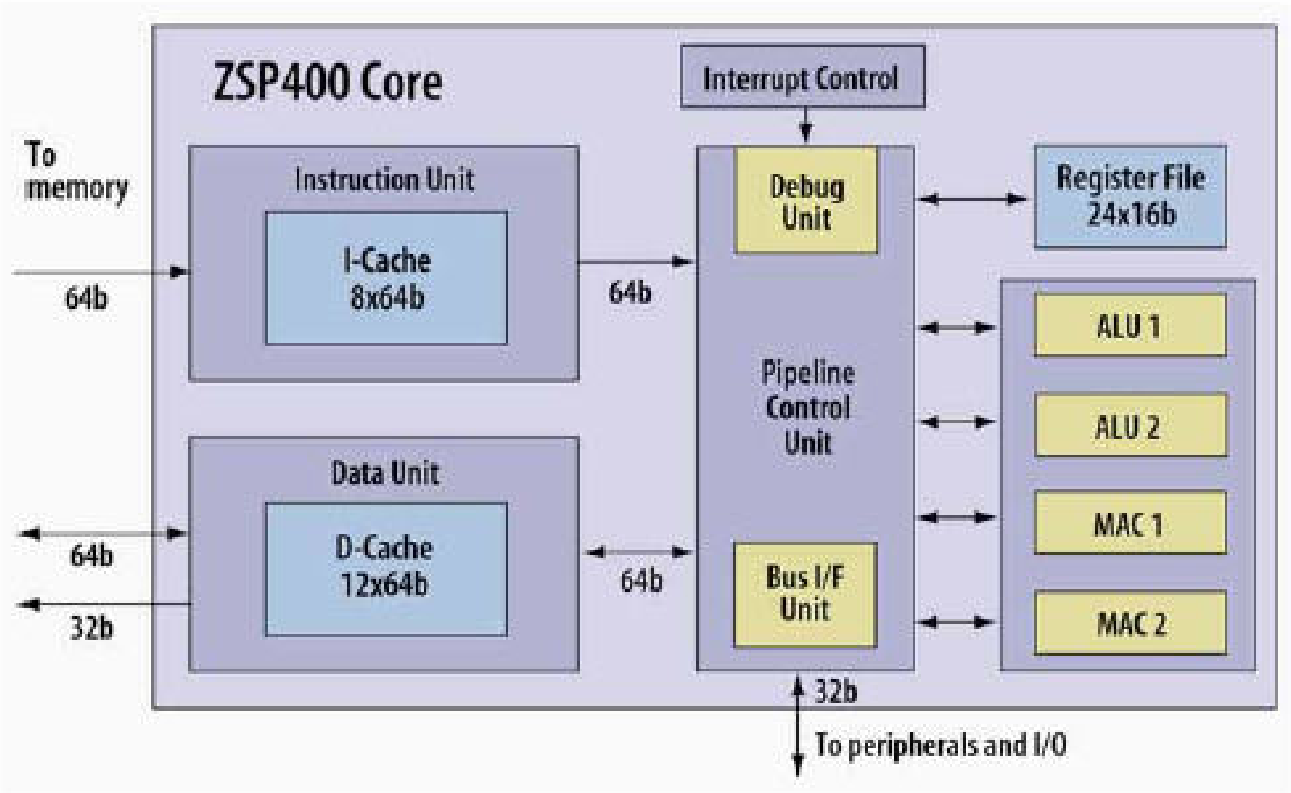
# CEVA-Teak (CEVA)

- Fixed-point DSP core
- Soft core (synthesizable)
- Computation unit
  + bit manipulation unit
- Two 16-bit x 16-bit multipliers
- Dual multiply-accumulate (MAC) in a single cycle
- 40-bit arithmetic/logic unit (ALU)

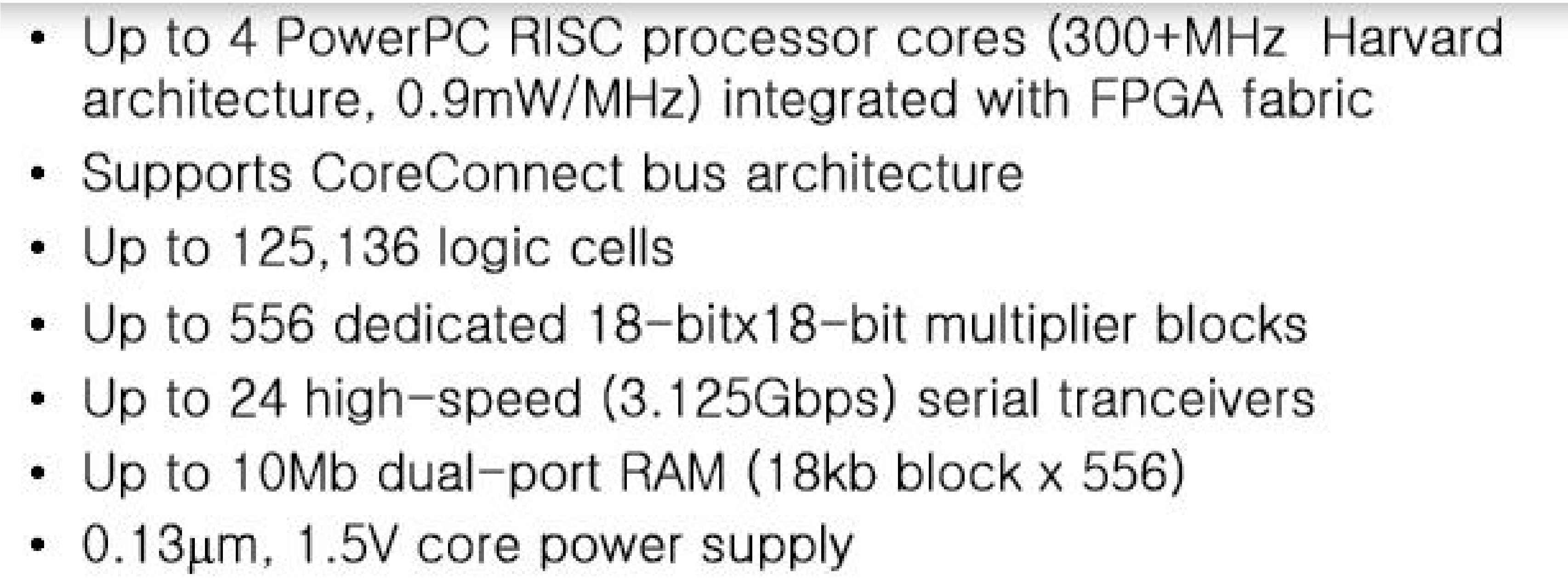


# ZSP400 (LSI Logic)

- Dual MAC/Dual ALU DSP core
- 4 issue superscalar 16-bit fixed point RISC architecture
- 5 nsec cycle time at 1.8V for 0.18 micron technology
- Synthesizable
- Two 16-bit ALUs in parallel or single 32-bit ALU



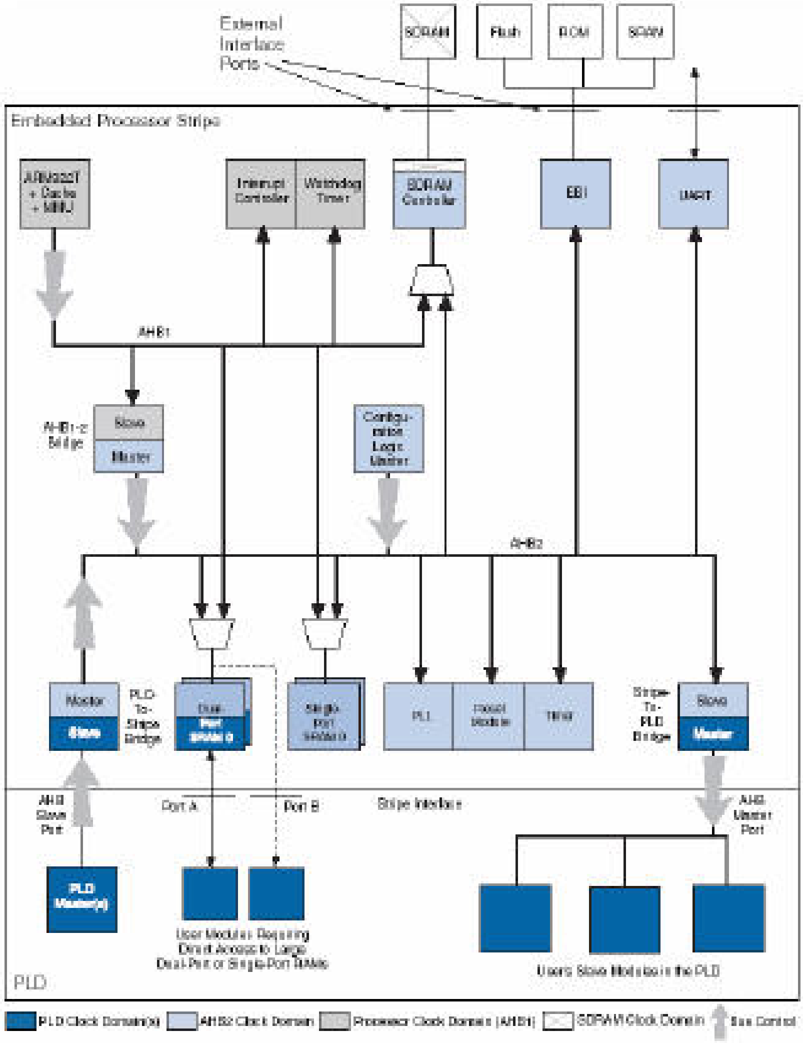
# Virtex-II Pro Platform FPGA (Xilinx)



- Up to 4 PowerPC RISC processor cores (300+MHz  Harvard architecture, 0.9mW/MHz) integrated with FPGA fabric
- Supports CoreConnect bus architecture
- Up to 125,136 logic cells
- Up to 556 dedicated 18-bitx18-bit multiplier blocks
- Up to 24 high-speed (3.125Gbps) serial tranceivers
- Up to 10Mb dual-port RAM (18kb block x 556)
- 0.13μm, 1.5V core power supply

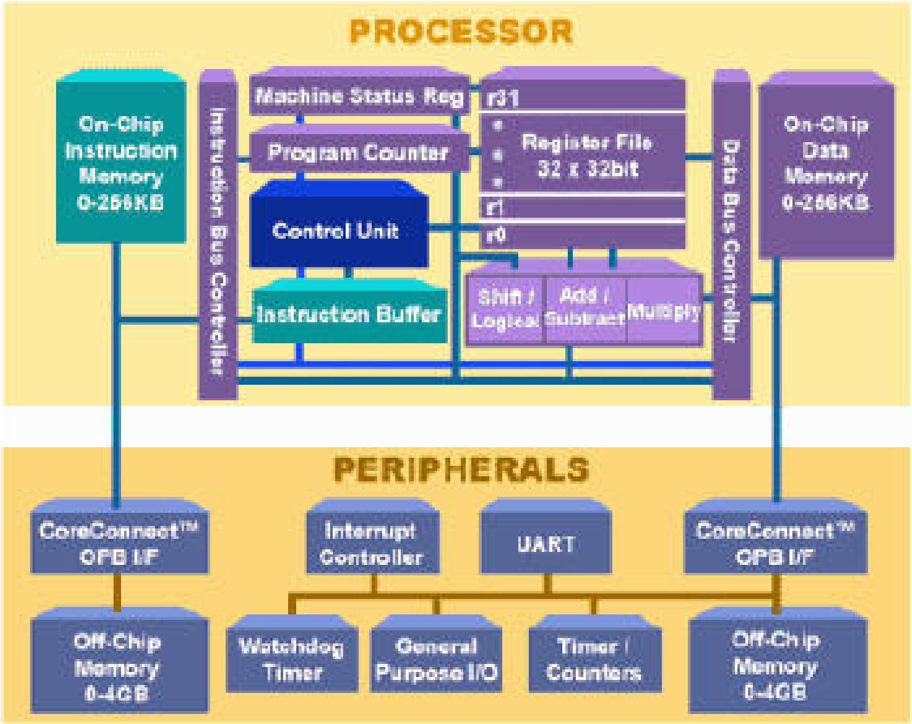
# Excalibur (Altera)

- ARM922T integrated with FPGA fabric
- Up to 200MHz
- Supports AMBA
- Up to 256K SRAM
- Up to 128K dual-port SRAM



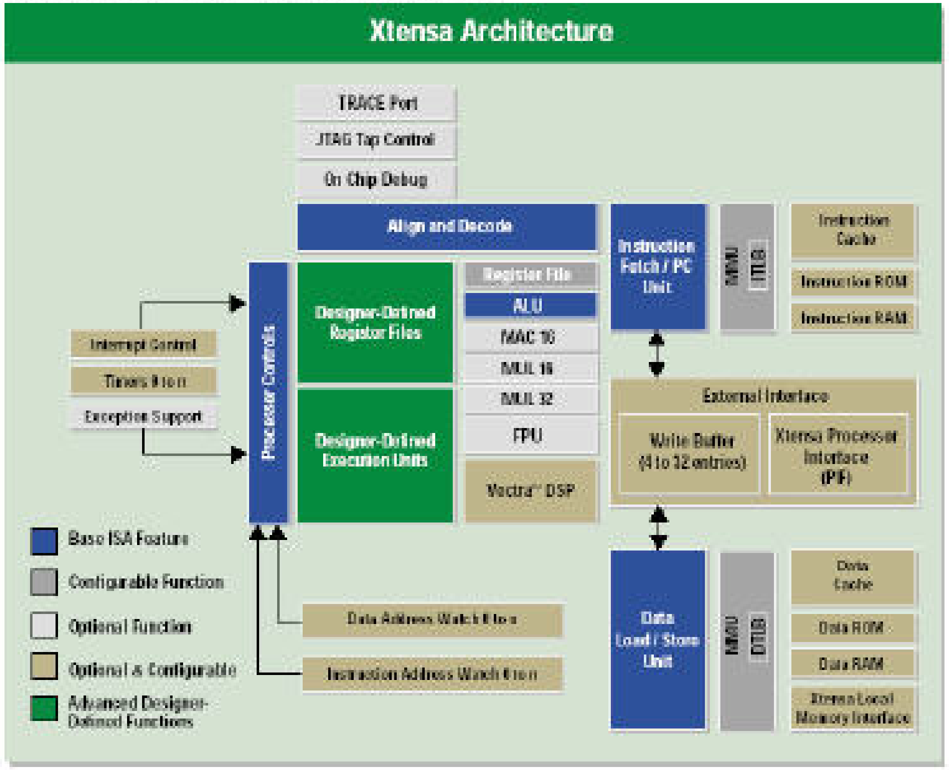
# MicroBlaze (Xilinx)

- Fixed processor cores given as soft cores
  - 32-bit RISC processor synthesized and programmed on a reconfigurable logic
  - Limited configuration can be done on peripherals and bus interfaces but the instruction set is fixed.



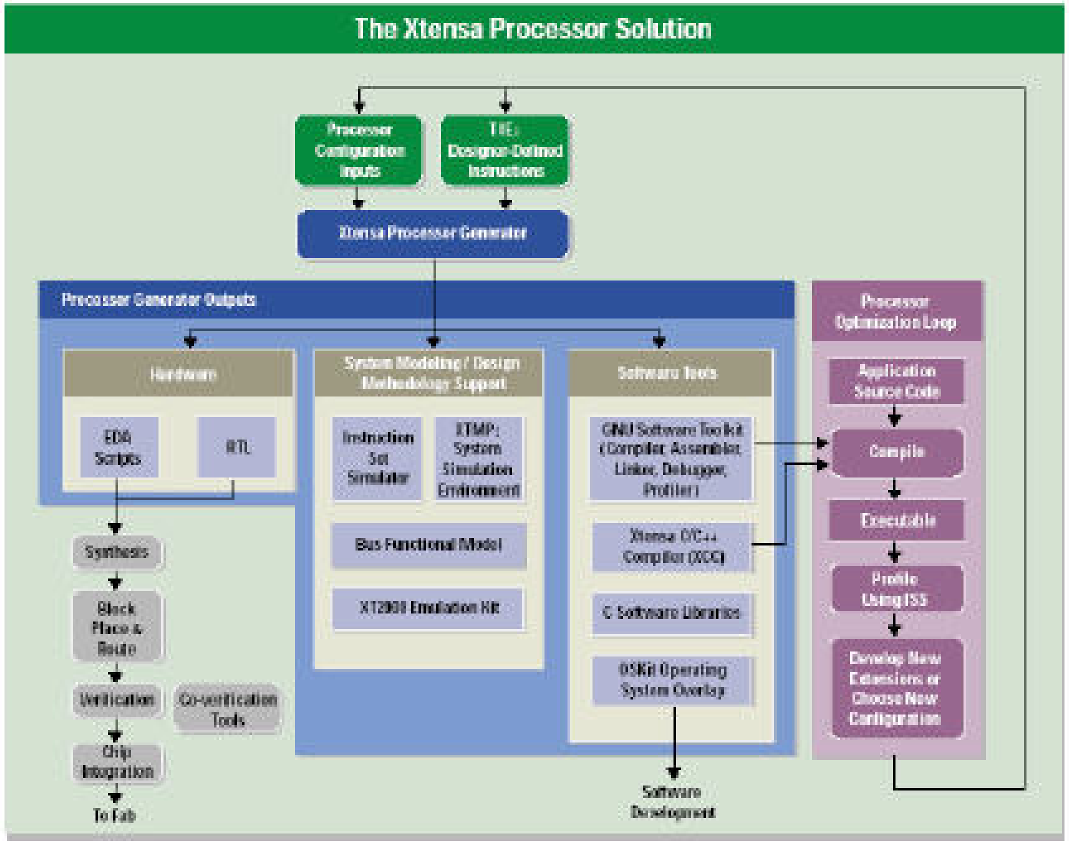
# Xtensa (Tensilica)

- Basic architecture
  - 24-bit instruction encoding
  - 4-bit register fields
  - Supports compound instructions
  - Zero-overhead loop
  - Limited instruction constant width



# Xtensa (Tensilica) (cont'd)

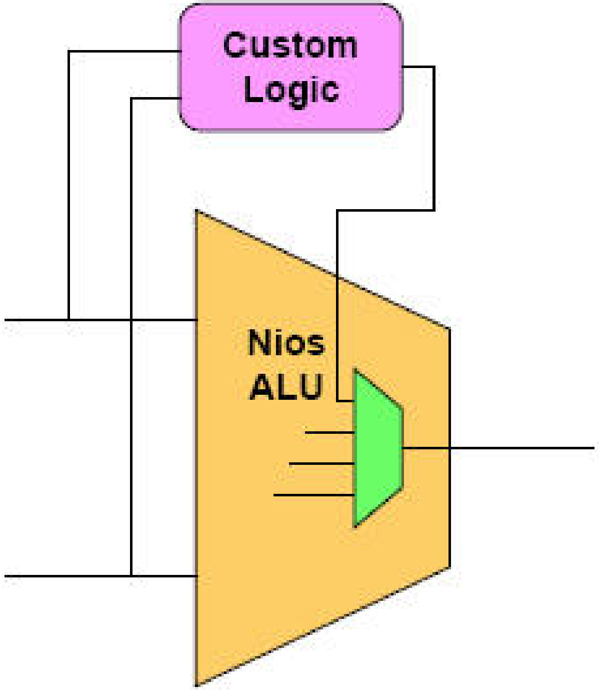
- Application-specific configuration using TIE (Tensilica Instruction Extension) language and methodology
- For consumer application, 23x improvement, 3.3x and 50x the performance of TM1300 and MIPS32, respectively



The Xtensa Processor Solution
Processor Configuration Inputs
TIE: Designer-Defined Instructions
Xtensa Processor Generator
Processor Generator Outputs
Hardware
EDA Scripts
RTL
System Modeling / Design Methodology Support
Instruction Set Simulator
XTMP: System Simulation Environment
Bus Functional Model
XT2000 Emulation Kit
Software Tools
GNU Software Tool kit (Compiler, Assembler, Linker, Debugger, Profiler)
Xtensa C/C++ Compiler (XCC)
C Software Libraries
OSKit Operating System Overlay
Processor Optimization Loop
Application Source Code
Compile
Executable
Profile Using ISS
Develop New Extensions or Choose New Configuration
Synthesis
Block Place & Route
Verification
Co-verification Tools
Chip Integration
To Fab
Software Development

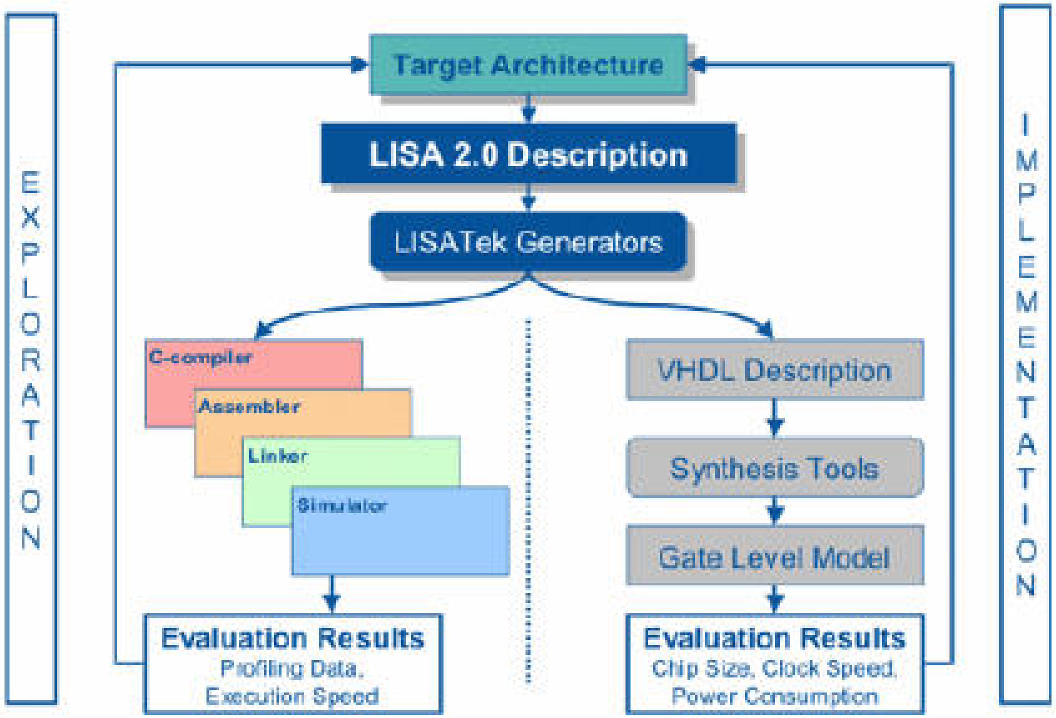
# NIOS (Altera)

- Configurable processor that can be programmed on a reconfigurable logic
- Only five (256 for NIOS II) opcodes can be used for user customizable instructions.



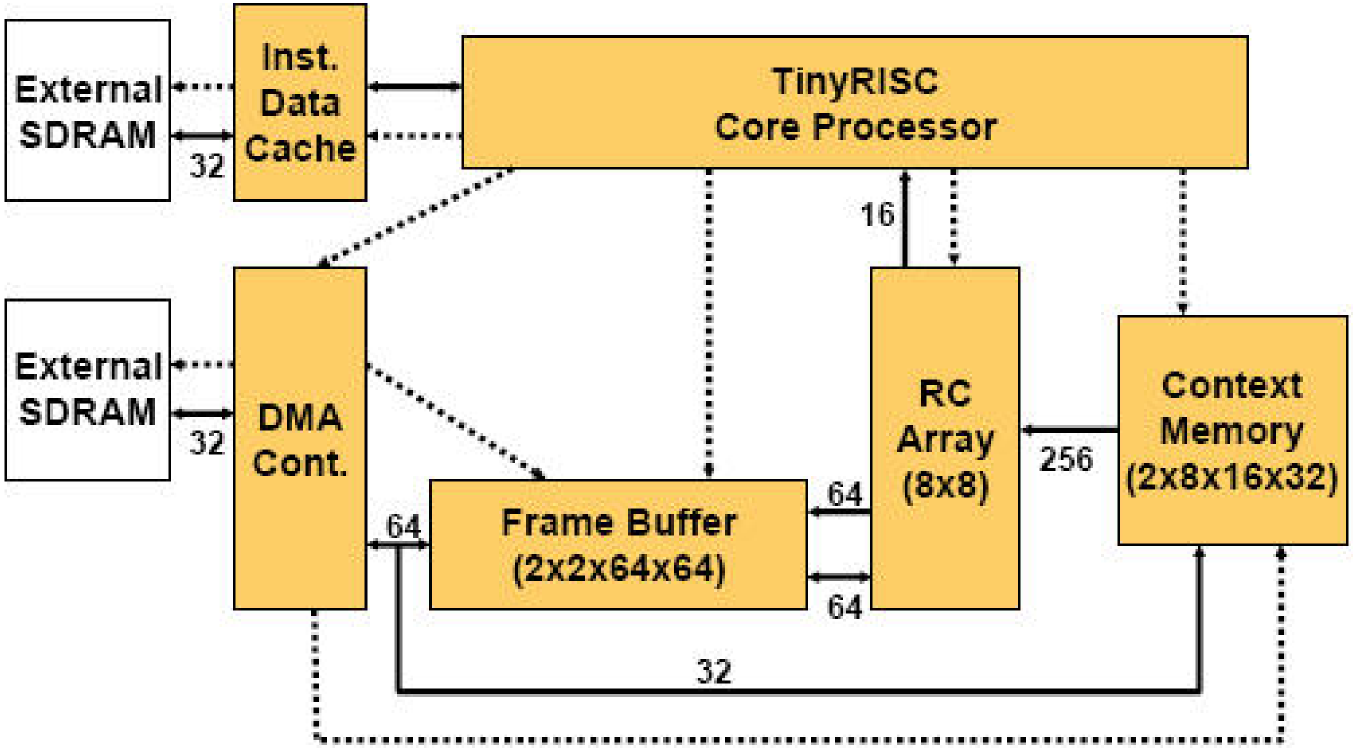
# LISATek (CoWare)

- Do not provide basic cores but start from scratch
- From an ADL (Architecture Description Language, LISA) description, they generate a compiler, a simulator, and a synthesizable HDL



# MorphoSys (Morpho Technologies)

- Fixed 32-bit processor core integrated with a coarse grain reconfigurable block


External SDRAM
32
Inst. Data Cache
TinyRISC Core Processor
16
External SDRAM
32
DMA Cont.
RC Array (8x8)
256
Context Memory (2x8x16x32)
64
64
Frame Buffer (2x2x64x64)
64
32