Winning With Innovation Platforms

Building Innovation Platforms
for
Consumer Electronic Product Development

Feb 28, 2012

IEEE Consumer Electronics Society

Santa Clara Valley Chapter

Gopi Kumar Bulusu

Sankhya Technologies Private Limited

www.sankhya.com

Agenda

- A Story of Long Term Growth
- Markets, Opportunities and The Consumer
  - Imperatives for Success
- Principles of System Level Design
  - System Level Design Works !
- Innovation Platforms for CE Device Development
  - Platform Elements
  - Design Flows
  - Example with SANKHYA Teraptor
  - Innovation Platform Maps
  - Emerge a Winner !

Story of Long Term Growth

- Companies in the space show consistent growth

Company	Market Cap Billion USD	Currency (Millions)	2000	2005	2010	2011
ARM	7.68	UK Pound	100	232.4	406.6	
Broadcom	16.33	US $	1100	2670	6818	
Texas Instruments	34.6	US $	9200	13392	13966	
Samsung	137	US $	8940	17210	27834	
Apple	365.83	US $	7980	13930	65220	108250

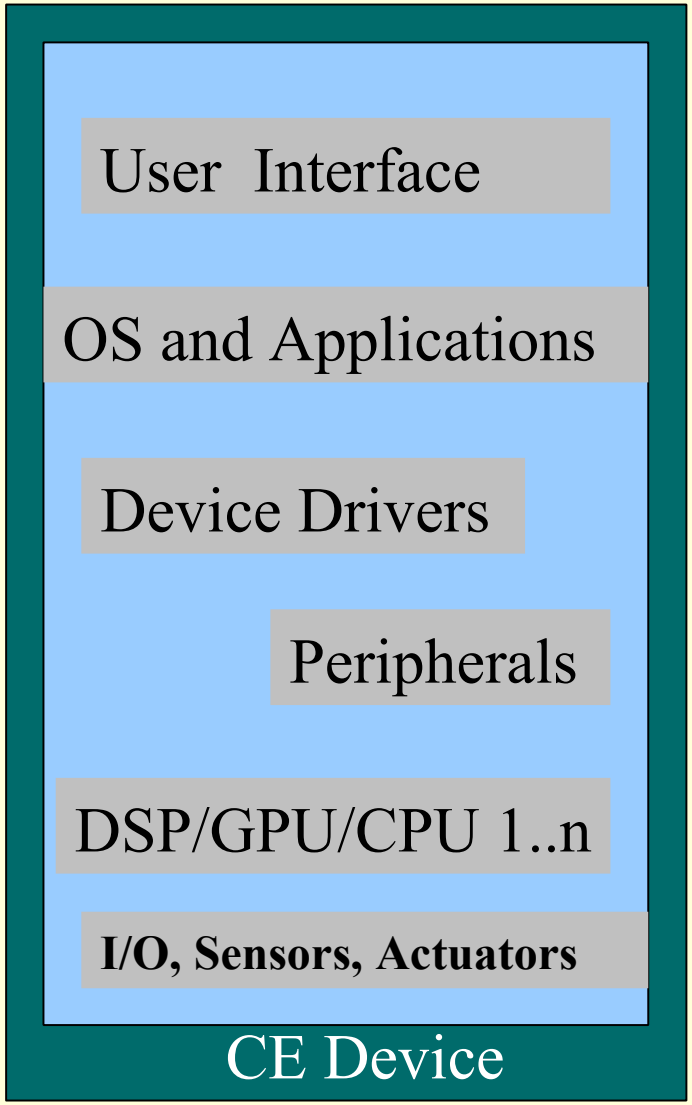
Disclaimer: Based on Information from Internet, For Illustration Purpose Only

A CE Device - CE 2012

Embedded System;
A computer embedded into a larger system or device

Generally Consumer Electronic devices are examples of embedded systems

SoC (System on Chip)
All components of an Embedded System designed on a Single Chip



Consumer Needs Drive The Market

- Consumer and Control Centric
  - Convergence
    - Multi-Function Devices
  - Divergence
    - Multiple Devices
- Information and Action Centric
  - Divergence
    - Local – NFC, Blue Tooth
    - Global
- Compute Centric
  - Convergence
    - Cloud

Convergence, Divergence and Emergence !

Consumers Learn and Want To Use Their Knowledge Even More.

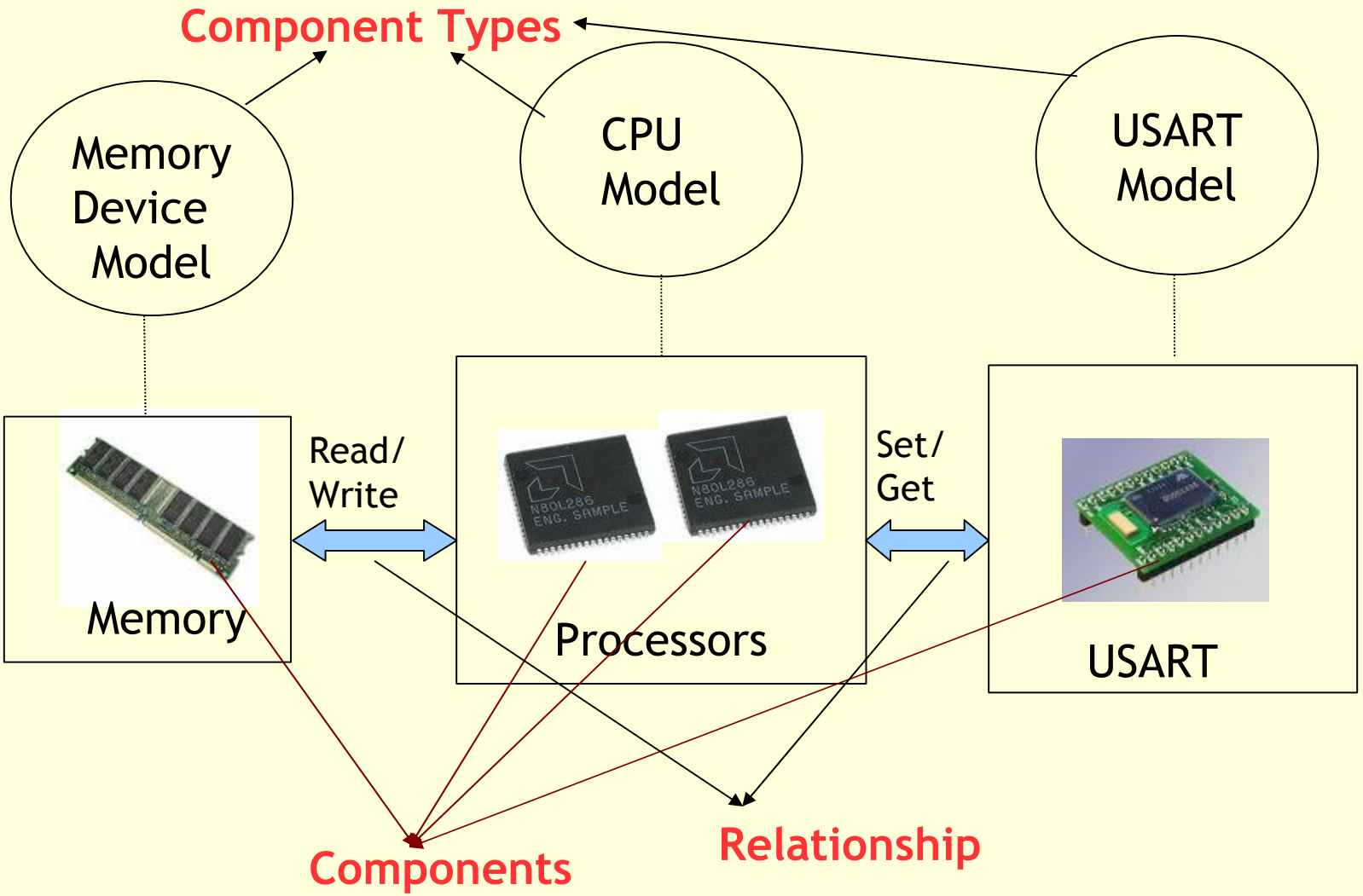
Succeeding In The Market Place

- Innovate : Deliver products that are
  - intuitive to use
  - have the right mix (convergence) of functions and performance empowering the consumer
  - at the right price point, globally
  - at the right time, ahead of competition
- Segment : Meet the needs of different consumer segments
  - Deliver multiple products and product variants (divergence)
  - Meet the needs of the global consumer
- Emerge(nt): Deliver products that are in-sync with consumer needs
  - Build core competencies and reusable IP
  - Create Innovation Platforms
    - Enable market driven specification, design, integration, verification and synthesis
    - Concurrent agile development

Model Driven, System Level

- Model Driven
  - Use abstraction. (Ex: Processor Model, System Model)
- Meta-Model Driven
  - Multi-level abstraction – Model for a Model
- Transparent Models
  - Models conforming to a meta-model
- Dimensions of Design and Development
  - Component Types, Simulation, Realization, Programming Tools (SDK), Intent Driven Verification
- Degrees of Freedom
  - Ability to modify the design in a particular dimension
    - Example: Processor core for specific application
- System Level – Architectural and Hierarchical System Models
- Multi-Dimensional Design Automation – Using a single model to automate multiple design and development activities

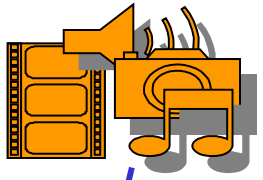
A System Model - Example



Principles of System Level Design / Benefits

Attribute	Benefit To Designer	Translating To Better Design	Competitive Advantage
System Level Design	Use Architectural Modeling Languages for Faster Development	Globally Optimal Designs Leading To High Performance, Low Power	Lower Design and Manufacturing Costs
Transparent Hierarchical Models	Multi-Dimensional Design Automation	A Comprehensive Solution	Higher Quality Low Support Costs
Architectural Models	Greater Large Block Design Reuse Across Projects	Improved Quality	Reduced Time To Market Segment with Product Variants

System Level Design Enables Innovation, Segmentation and Emergence

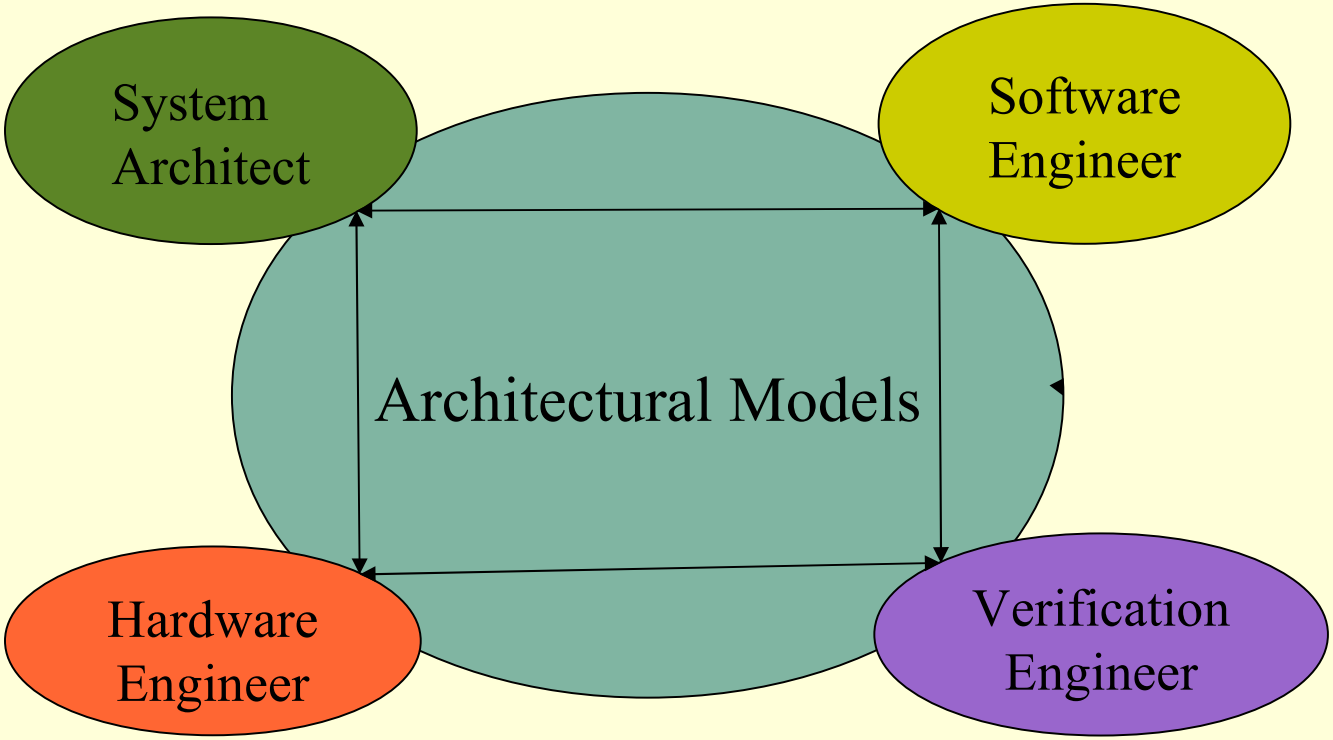


Innovation Platforms For CE Design and Development

Platform Elements (partial list)

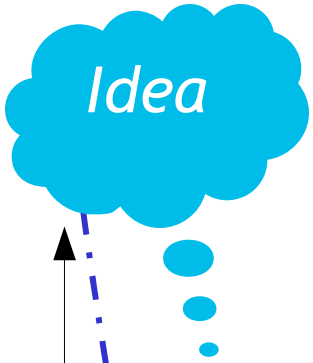
- **Languages and Models**
  - **Architectural System Modeling, Timing Models**
  - **Processor Architecture (ISA) Modeling, Timing Models**
  - **Hardware Behavioral Modeling**
  - **Behavior Modeling (Software, Hardware Emulation)**
  - **HMI / User Interface Modeling**
  - **IDE**
- **Dimensions of Modeling, Design and Automation (Tools)**
  - **Simulation: Component, System, Processor; Functional, Timing, Performance**
  - **Verification Synthesis: Processor, System**
  - **Debugger**
  - **SDK Synthesis / Meta-SDK: Assembler, Linker, Code Generator**
  - **HLS / VHLS : Architectural to Synthesizable RTL**
  - **Communication and Bus Synthesis**
  - **Partitioning and Scheduling**
  - **C/C++ Compiler Synthesis / Meta-Compilers**
  - **Driver (HAL) and PAL Synthesis**
- **Degrees of Design Freedom (Transparent Models)**
  - **System Model, Processor Models, Components, System Software, Application Software, HMI Models**
- **Library of Components**
  - **Processor Models, Memory and Cache Models, Serial and Parallel Devices**

Platform => Agile Concurrent Development

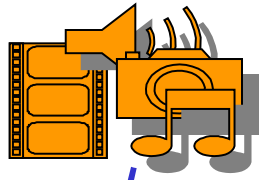


Concurrent Development Activities

- Solution Specialist: Market Research and Product Specification
- System Architect: Product Prototyping, UI and UE and Analysis
- Processor and Tools Architect: Processor Modeling and Software Development Tools Like Compilers, Assemblers, Linkers, Debuggers, Simulators
- Peripheral Designer : Peripheral Modeling USB, Ethernet, Bluetooth, CAN, RFID, GPS, GSM, Sensors – Behavior, Hardware
- System Software Architect: OS and Driver Development / Porting ; Linux, Android Etc.
- Media Architect: DSP, Audio/Video Codecs
- Hardware/ Prototype Designer : FPGAs/ASICs
- Application Architect: UML, Developing UI, MVC, Web
- Circuits and Circuit Board Design: PCB Design, DFM
- Physical Design: “Crash Testing” ...



System Level Design Flows



Market

Solution

STB, DVD Player, Camera, MP3

System Models

Processor, Perihperal Models

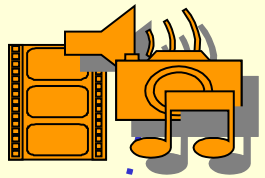
Technology

Design Abstraction

Product Abstraction

Embedded System Level Design

Application Driven Embedded System Level Design
For Consumer Electronic Product Development
Specify => Model => Integrate
Virtualize => Verify => Synthesize



Faster to market translates to higher price points and greater market share.

Idea

Market Driven Application Specification

Off-The-Shelf IP
Hardware and Software
Components

Reusable HW/SW Developed In Parallel Using Virtual Prototypes

Reduced TIME TO MARKET

Very High Level Synthesis
Custom Synthesis Tools

Automatic component reuse spanning application, system, peripherals and processors

System Models (SSDL)
Teraptor Designer

Transparent Modeling Languages for Processor and System Design

Virtual Prototypes
With Teraptor Player

Processor Models (SMDL)
Teraptor MSE, SDK, Debugger
Model Export, Synthesis

Model Driven Tools For Managing Design Tradeoffs

Teraptor ™
Eclipse Based IDE
MSE and Verifier
SMET for LLVM
Very High Level Synthesis Debugger
SDK
Player
Channels
C Compiler SA*
C Compiler HPC*

3- 9 Months !

Innovation Platform Map

1. Languages

SNO	Layer	Model-Type	Modeling Language
1	System	Architectural ER Model	SSDL (or SysML)
2	Component	Processor ISA	SMDL
3	Component	Hardware Emulation	C++

2. Dimensions of Automation and Tools

...

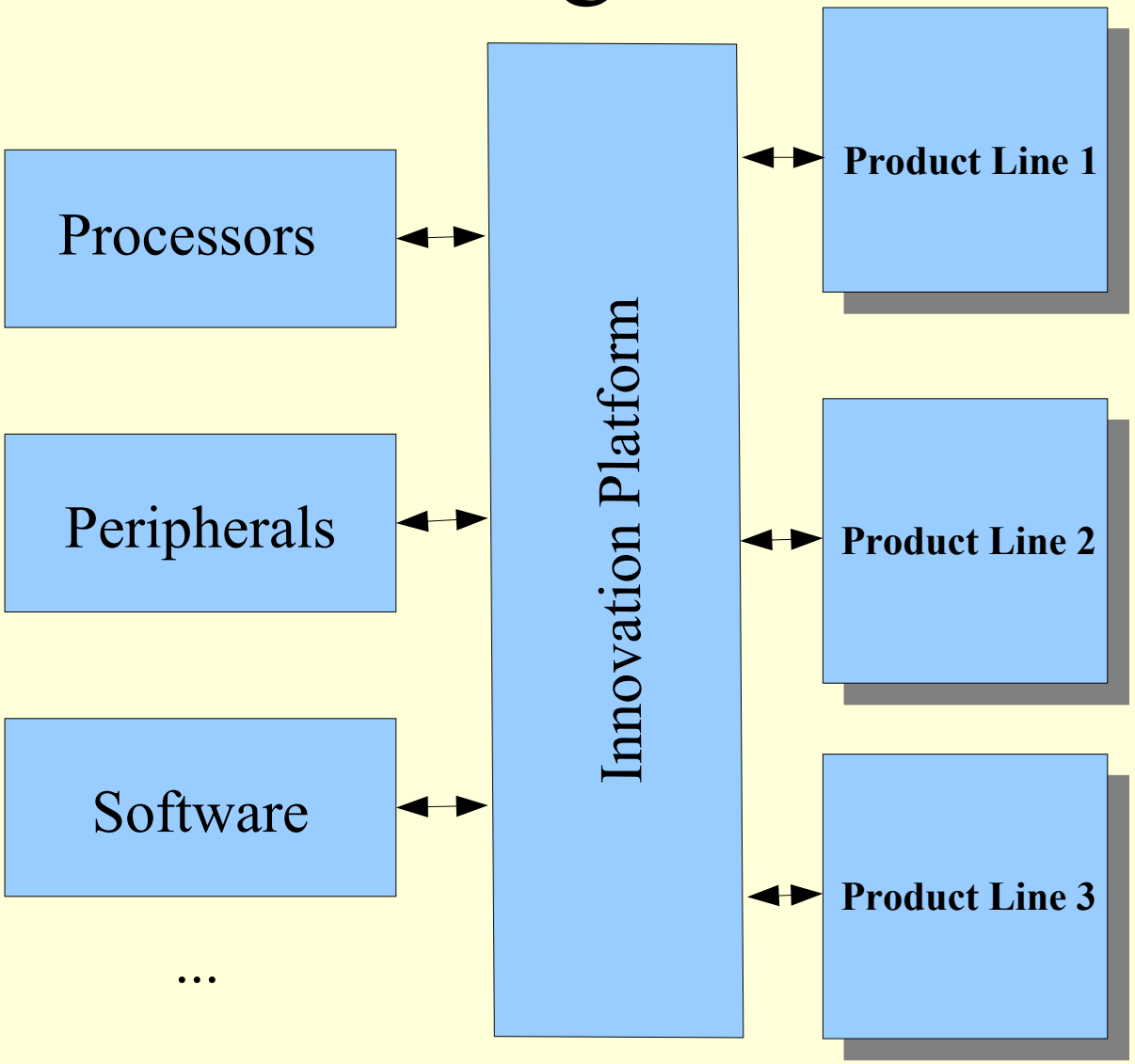
3. Degrees of Design Freedom (Transparent Hierarchical Models)

...

4.Library of Components

...

Emerge a Winner !



Innovation Platforms for CE Design Help Your Organization To Win In The Market Place By Practicing System Level Design and Thereby:
+ Supporting Concurreng Agile Development
+ Enabling Globally Optimal Designs
+Allowing Greater Design Reuse Across Product Lines
+ Ensuring Multi-Dimensional Design Automation

Questions ?

- Innovation is the specific instrument of entrepreneurship... the act that endows resources with a new capacity to create wealth. – Peter Drucker

E-mail ***gopi@sankhya.com***

http://www.sankhya.com/contact.html

http://www.sankhya.com/info/products/teraptor/teraptor.html

Chennai

Sankhya Technologies Private Limited
"Jayashree", Third Floor,
#13/2, First Street, Jayalakshmipuram,
Nungambakkam,
Chennai 600 034 INDIA
Tel: +91 44 2822 7358
Fax: +91 44 2822 7357

Visakhapatnam

Sankhya Technologies Private Limited
Sankhya Rishikonda Development Center
#10 and 11, Hill 2, Rishikonda IT SEZ
Gitam Post, Madhurawada
Visakhapatnam 530 035 INDIA
Tel: +91 891 646 2663
Tel/Fax: +91 891 273 7270

World Wide Sales

E-mail: sales @ sankhya.com

US Sales

Sankhya Technologies Inc.
Tel: +1 408 556 9757

Support

E-Mail:support@sankhya.com

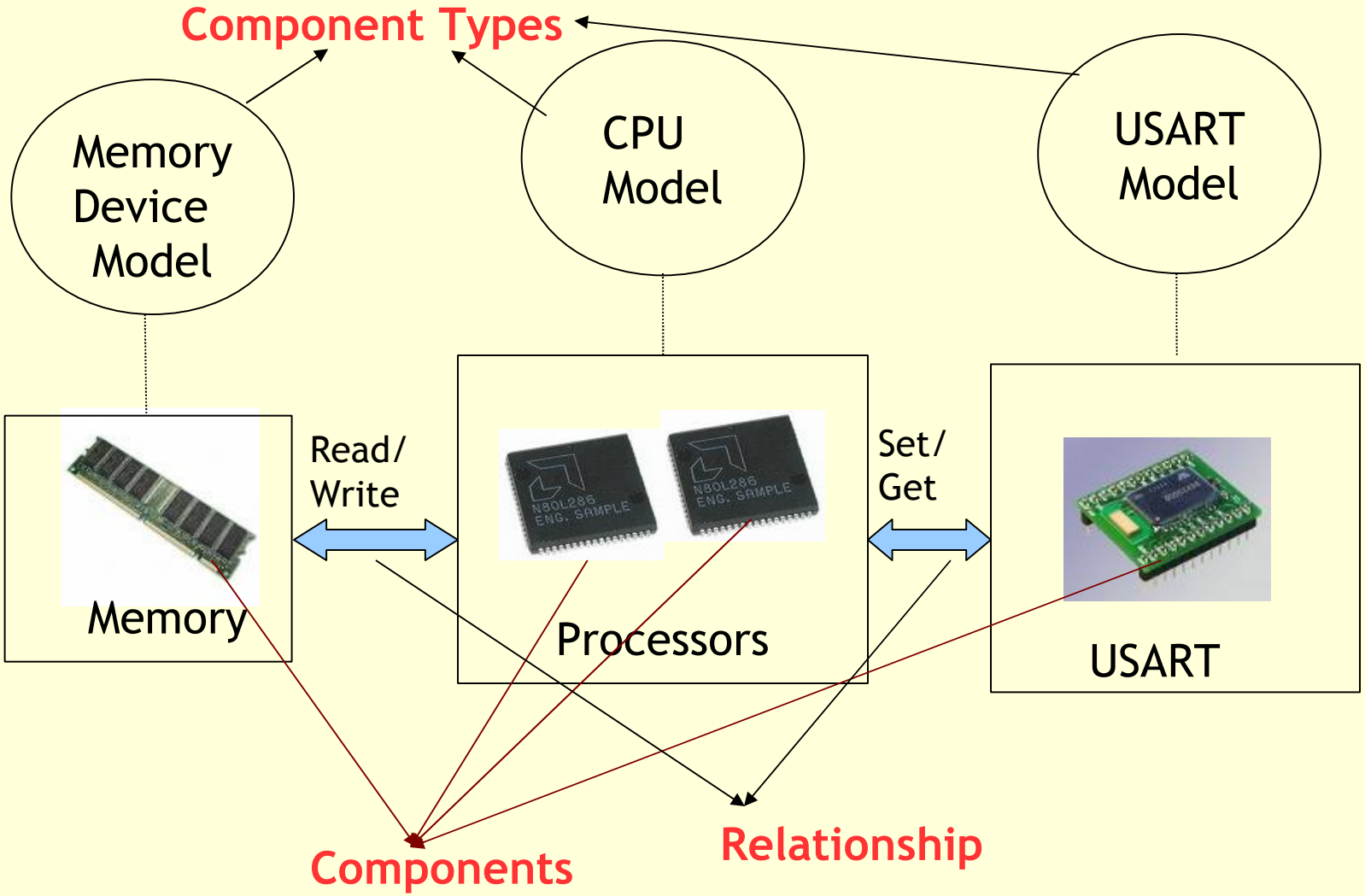
Corporate Training

E-mail: education @ sankhya.com

Consulting and Offshore Development

E-mail: scsinfo@sankhya.com

System Model – A Quick Example



```
model processor DLX dlx.md
model device SRAM SRAM.dll
model device USART USART.dll
```

Component Type Definitions

```
component DLX cpu1
component DLX cpu2
component SRAM ram size=0x2000 mode=rw
component USART usart
```

Component Definitions

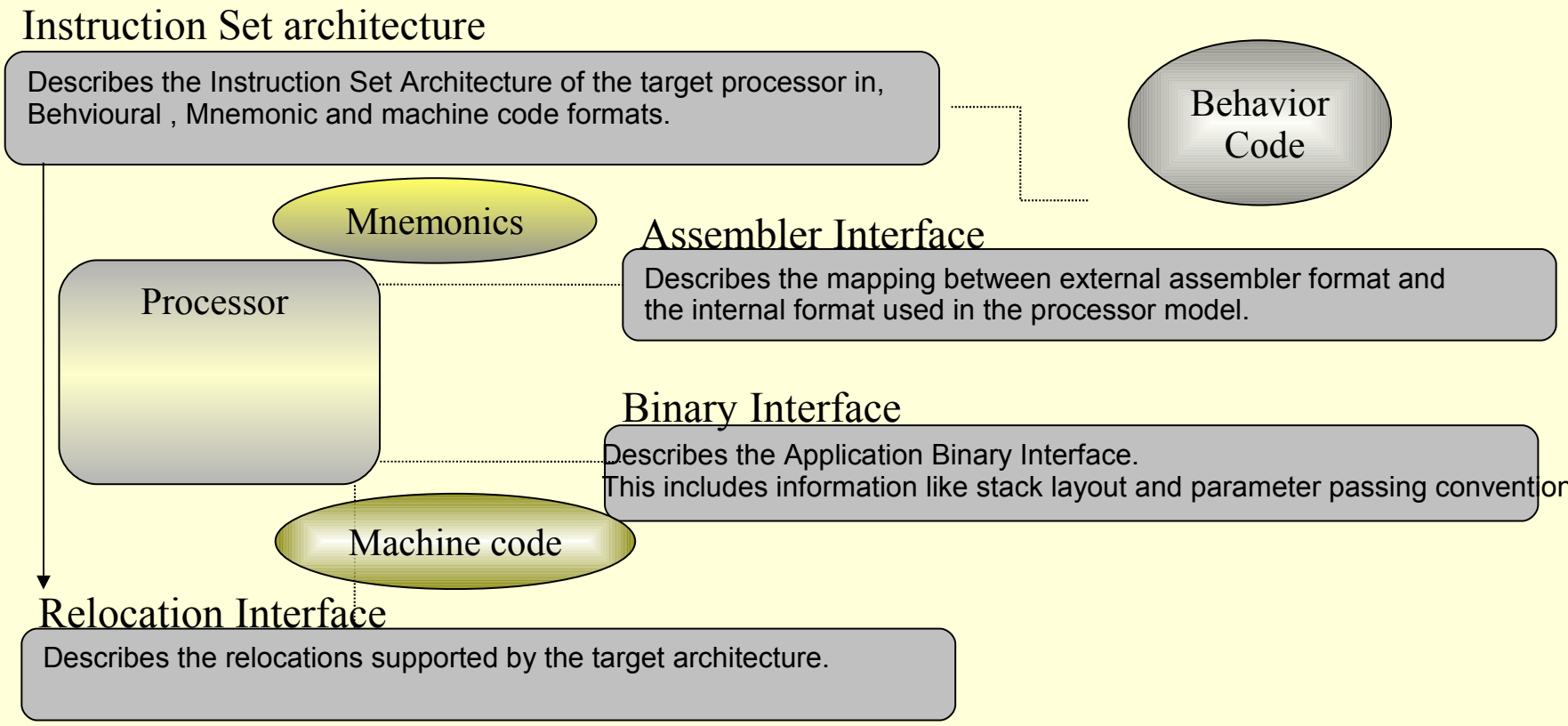
Attribute

```
map memory cpu1:address.0x0 ram:STDRAM
map memory cpu1:address.0xffff0000 usart:tx_register
```

Relationship Definitions

Resource

- Specify several views of the processor like, Architectural view, Mnemonic View and Machine code



SMDL Example – Operands

- Immediate Operand

```
// 5-bit unsigned immediate value
Operand o_uimm5 : Integer {
        range (i = 0, 31, 1) {
                cbcode = { "$i" };
                asm   = { "$i" };
                mcode = { "$i" };
};
};
```

- Register Operand

```
// 32 general purpose registers + 3 special purpose registers
Operand o_gprv : Register {
        range (i = 0, 34, 1) {
                cbcode = { "r$i" };
                asm = {"r$i"};
                mcode = {"$i"};
        };
};
```

SMDL Example – Addressing Modes

- E.g, PC Relative Offset

```
Operand o_i_pcrel {
        o_imm16 o;
        cbcode = { "+", "pc", "<<", o, "2" };
        asm = { expr("<< $o 2") };
        mcode = { o(0,16) };
};
```

SMDL Example - Instruction

- Here's an Example of SMDL Representation of a simple *'add'* instruction of the **DLX processor**

Mnemonic	**Opcode**	**Operands**
add rd rs1 rs2	000001	rs1 rs2 rd 00000 00000

```
Instruction i_rrr_add {
    o_gprv rd, rs, rt;

    cbcode = { "{width=32}=", rd, "{sign=u}+", rs, rt };

    asm = { "add", rd, rs, rt };

    mcode = { or(0,7,0b00000000); or(7,4,0001); rd(11,5); rt(16,5);
    rs(21,5);or(26,6,0b000001)};

};
```

Common Behavior Code

Mnemonic

Machine Code

SMDL Example – Instruction Bundle

- Allows VLIW type instructions to be modeled.
- Example: A 128-bit VLIW

```
Bundle b_mii {
      i_mem i1;
      i_alu i2, i3;

     b_width = 128;
     cbcode = { i1, i2, i3 };
     asm   = { i1, i2, i3 };
     mcode = { i1(0,48); i2(48,40); i3(88,40)  };
  };
```

SMDL Example – Instruction Attributes

- **Specify width as 32 bits for this instruction.**

```
cbcode = { "{width=32}=", rd, "{sign=u}+", rs, rt };
```

- **Cycle Attribute - specifies number of clock-cycles to be used.**

```
{cycle=2}++ {read-cycle=1,write-cycle=3}r1
```

- **Effects Attribute – Specifies side-effects of a cbcode operation**

E.g. Subtract with Carry (sbc) instruction for two 32-bit values

```
// SBC{cond}{S} Rd, Rn, <Op2>
   Instruction i_rro_sbc {
   o_gprv rn, rd;
   o_operand2 op2;
   cbcode = { "=", rd, "-", "{!effects=1,
                              !effects-create-mask=NZCVttttttttttttttttttttttttttt,
                              !effects-destination=flags}-", "-", rn, op2, "bf", "29", "1",
              "flags"
            };
   asm   = { "sbc", rd, rn, op2 };
   mcode = { or(26, 2, 0b00); or(21, 4, 0b0110);rn(16, 4); rd(12, 4); op2(0, 12) };
   };
```

N, Z, C, V, t – Negative, Zero, Carry, Overflow, Transparent (not affected)
flags – Status register

SMDL – Key Features

- The SMDL model contains sufficient information to automate various tasks like CPU design, verification,system and tool development, code generation etc.

- Extensible Language :
  Rich Set of Language attributes enables to Model
  - Cycle- Accurate Processors
  - Complex Pipelining Architectures

- Common Behavior Code – The H/w - S/w Unifier.