



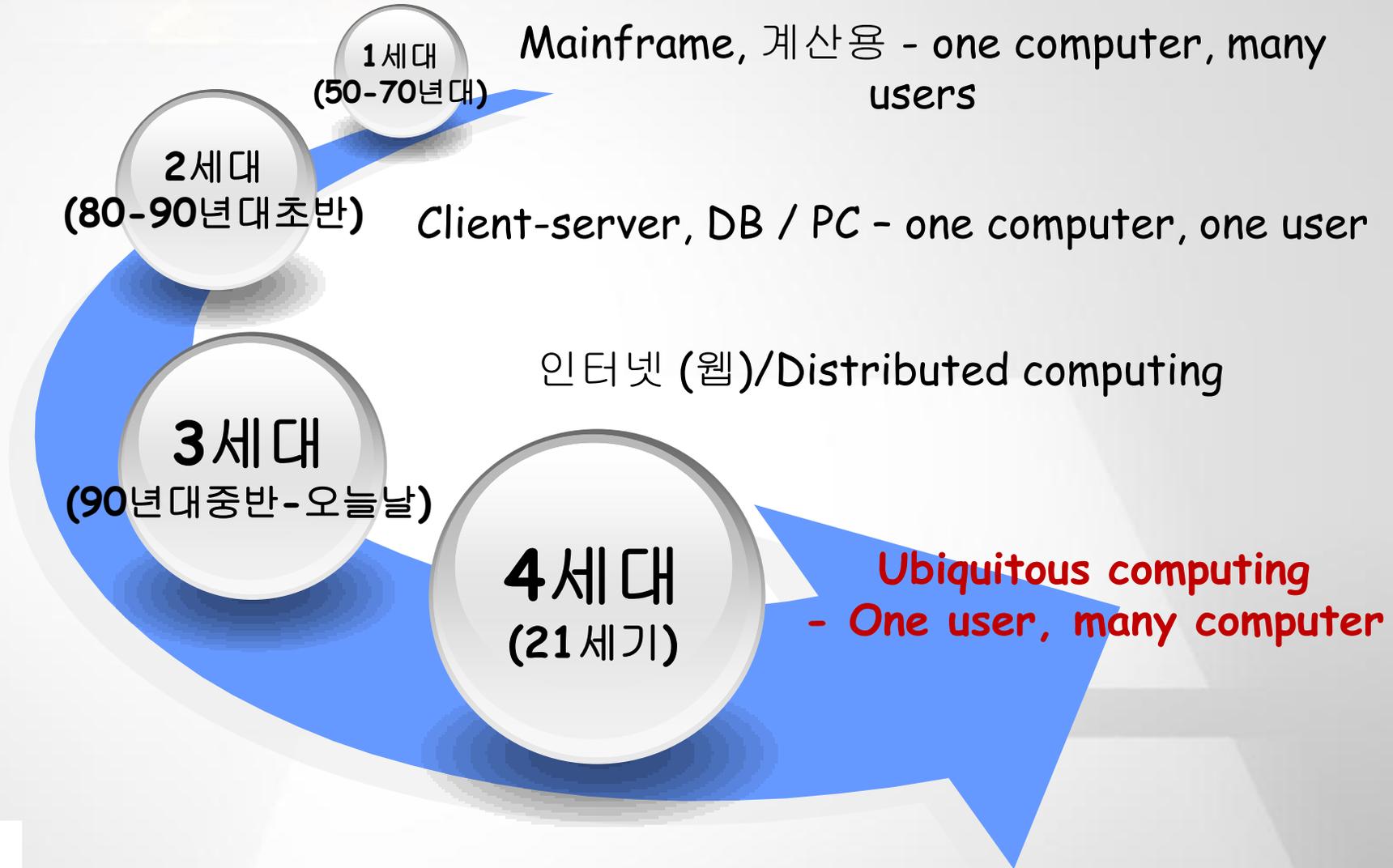
차세대 컴퓨터 시스템 설계 기술

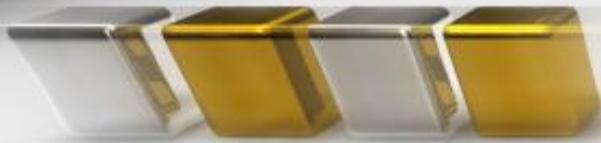
Feb. 15, 2013

서울대학교 컴퓨터공학부
하순희 교수

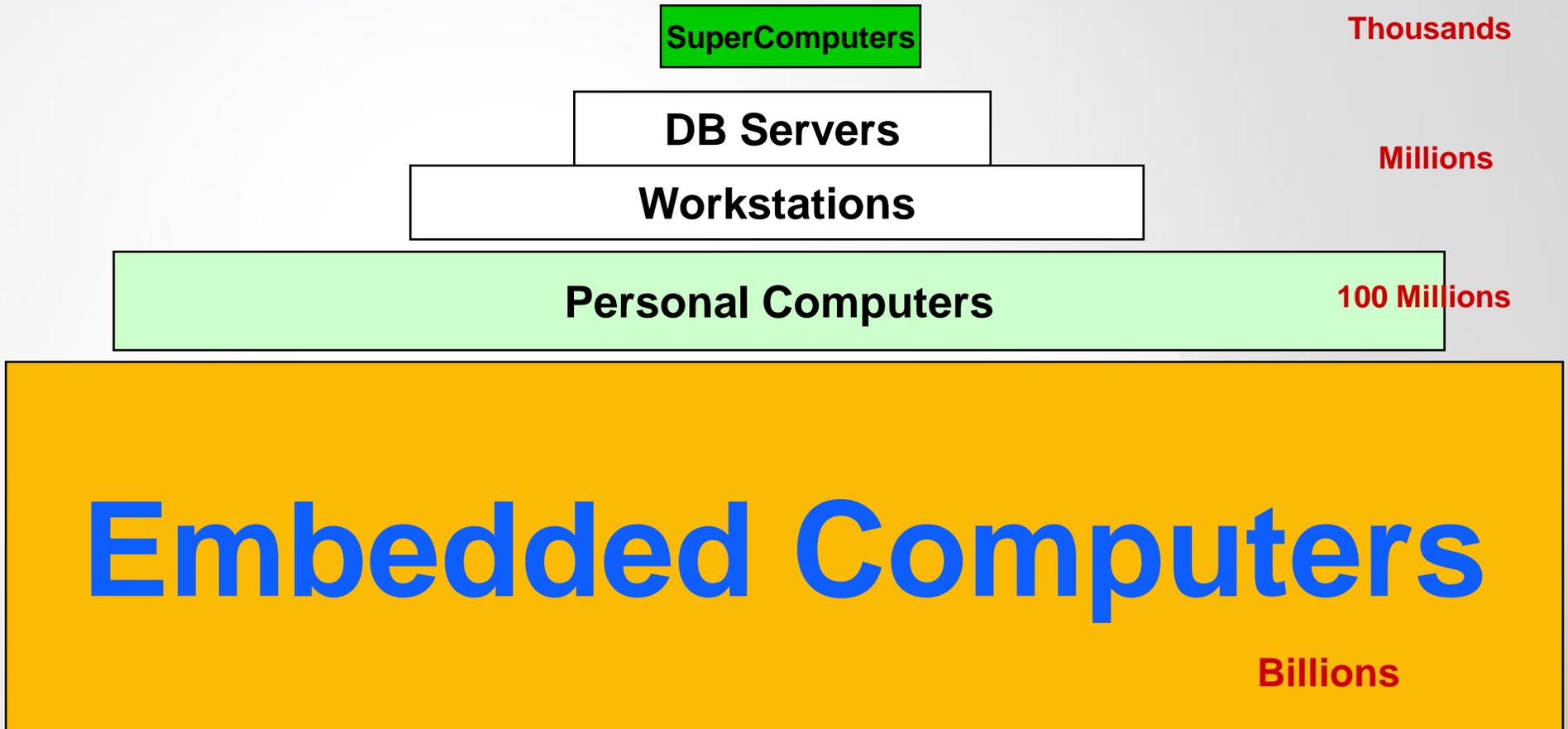


컴퓨팅 패러다임의 변화





컴퓨터 플랫폼의 피라미드



임베디드 컴퓨터

☞ Computers that are interacting with the environment (physical systems)

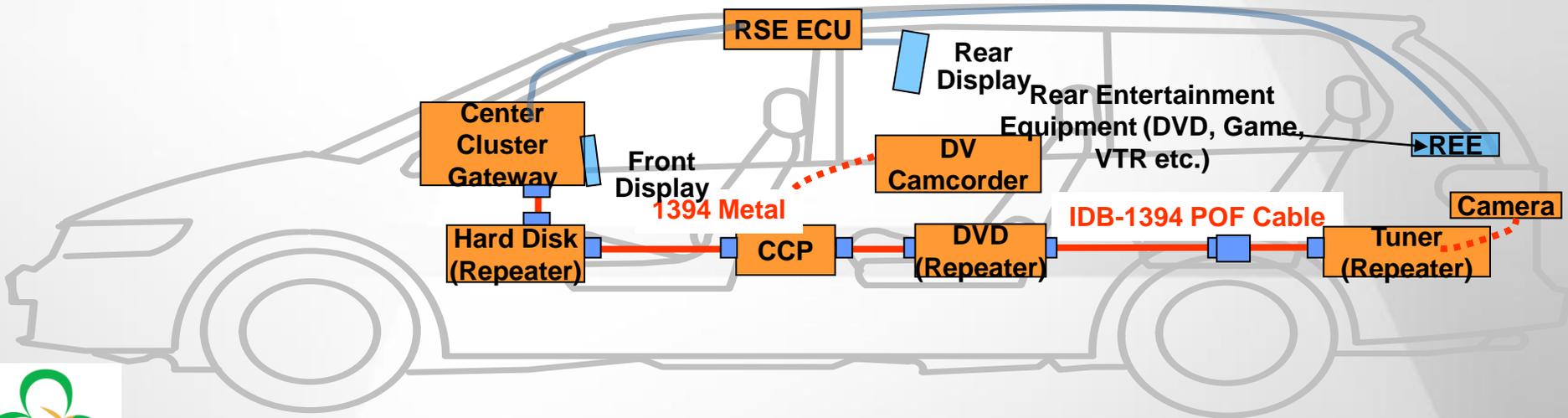
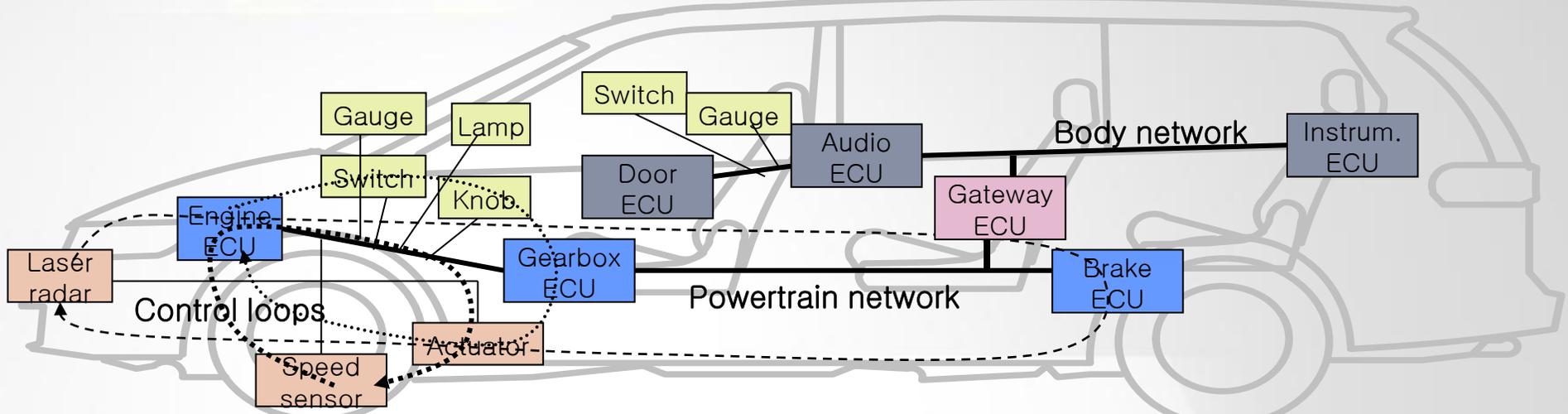
- Application specific system

☞ 예시

- Automotive controllers
- Avionics
- Medical devices
- Multimedia devices
- Industrial controllers
- 그외 다수



Automobile is an Embedded System





Embedded Systems 의 특징

- ❏ 시간 제약 조건 (*Real-time constraints*)
 - Transformational/ Interactive/ Reactive
- ❏ 이종 병렬 시스템 (*Heterogeneous parallel computing system*)
 - CPUs (multi-tasking SW) + Hardware
- ❏ 정확한 설계가 무엇보다 중요함
 - On-line debugging is often impossible
- ❏ 요구사항
 - Cost-performance ratio
 - Power/energy consumption
 - User friendliness
 - Shape and Weight
 - Time to market
 - reliability, safety, availability



컴퓨터 시스템의 설계

❏ Programmer vs Architect

- Programmer is not a computer engineer!

❏ We need "true" engineers to build complex computing systems

- Follower vs pioneer



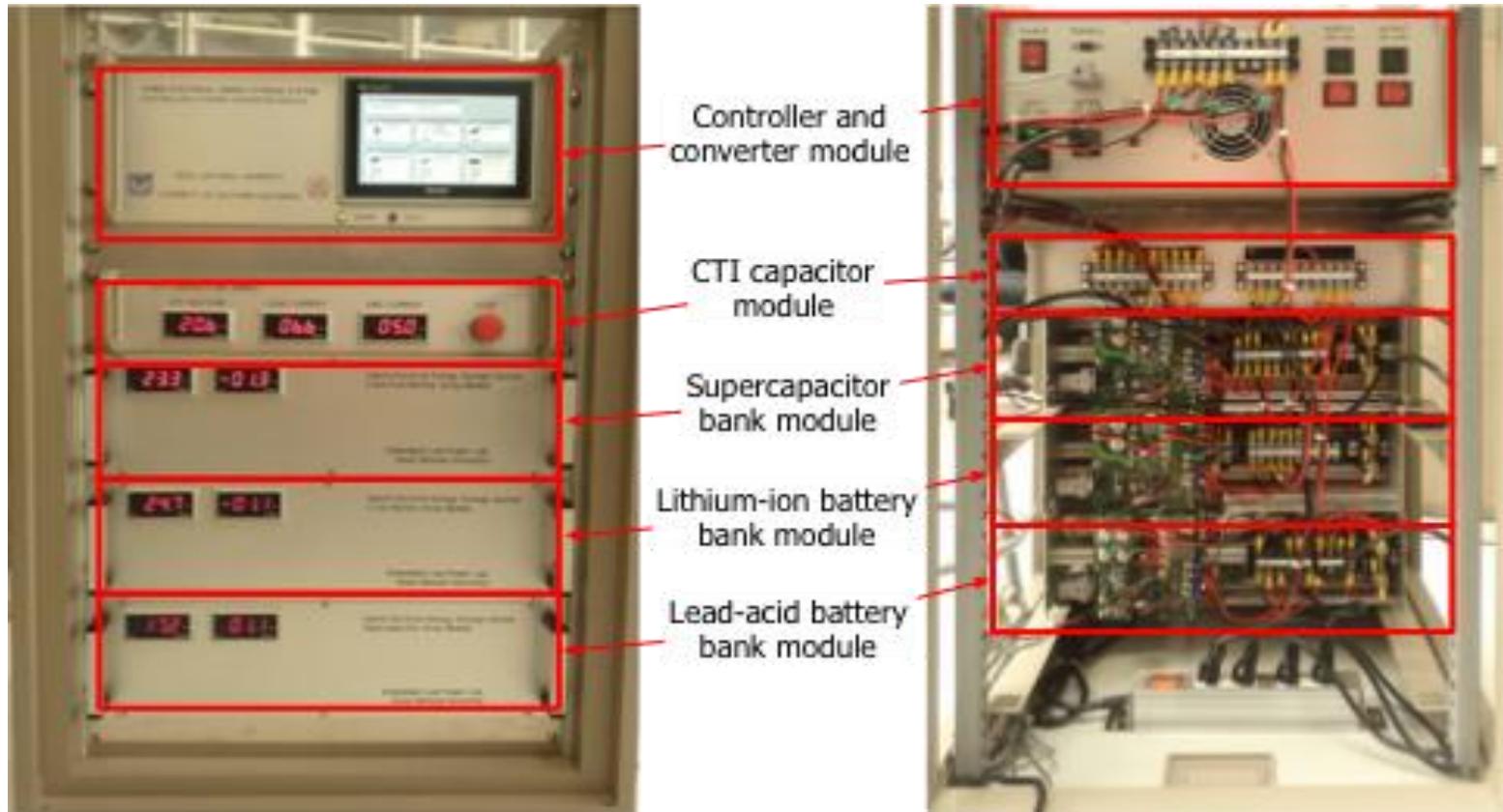
장래혁 교수: 컴퓨터하드웨어 응용 시스템을 설계 및 구현

저전력 시스템, 에너지 시스템, 지진계, 무인항공기 등



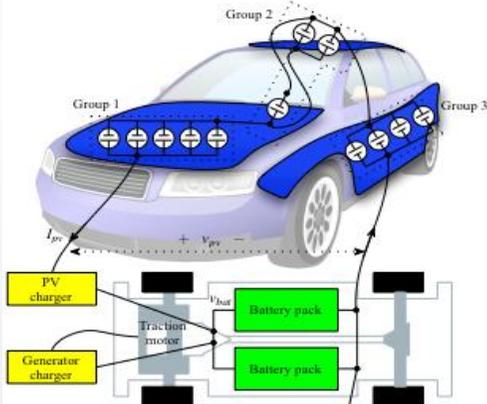
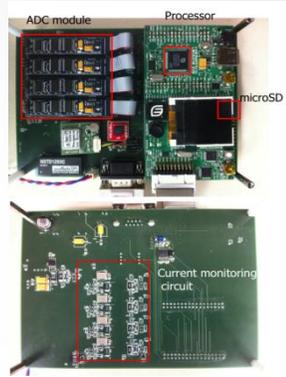
전력 수급 조절을 위한 전기에너지 저장장치

- ❏ 전기에너지 저장장치를 컴퓨터시스템 설계 개념을 도입하여 성능을 획기적으로 개선함



다양한 융복합 연구

☪ 컴퓨터 시스템 설계 및 설계 자동화 기법을 인접 학문분야에



태양전지, 연료전지의 최적화, 하이브리드 자동차에 적용하여 연비를 높이는 연구



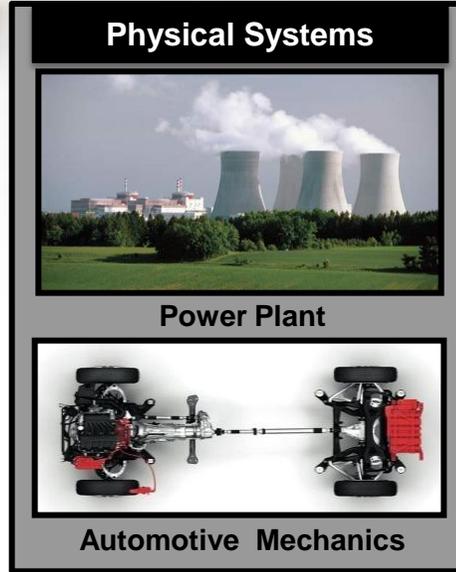
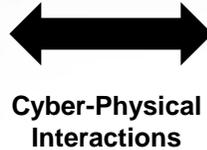
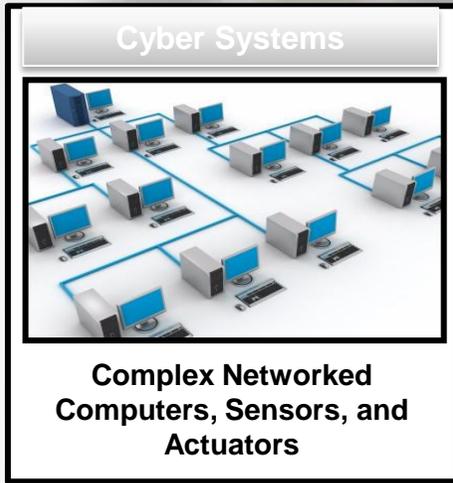
컴퓨터 시스템의 저전력 설계를 통한 배터리 수명 증대를 해저 지진계에 적용하여 심해에서 오랜시간 동안 작동하게 하는 연구



컴퓨터 시스템의 저전력 설계를 통한 배터리 수명 증대를 무인항공기에 적용하여 비행시간을 늘리는 연구



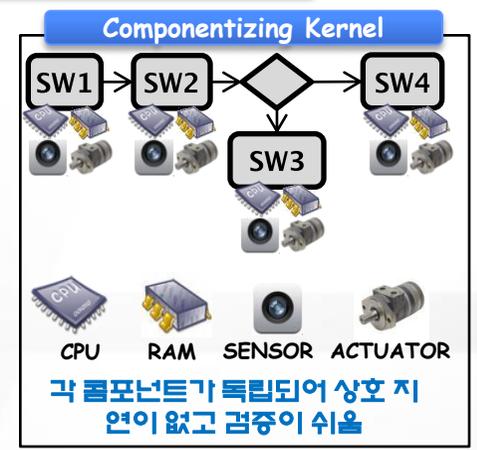
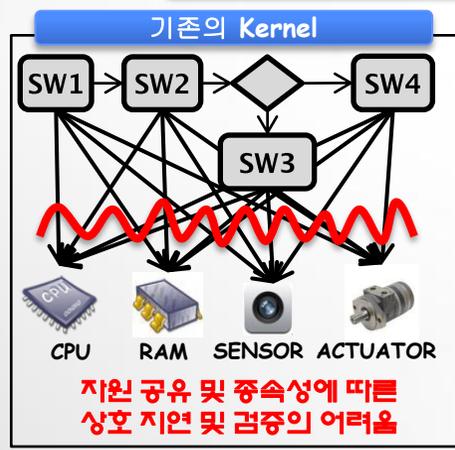
이창건 교수님 연구 분야



CPS?

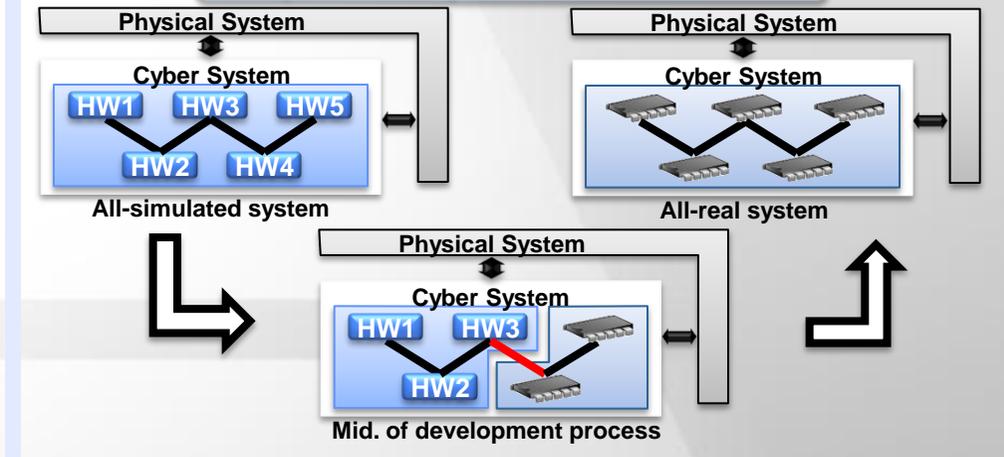
- 물리적 환경에 내재된 네트워크와 컴퓨터들(Cyber-systems)이 물리적 프로세스(Physical-systems)와의 상호 교류(feedback loops)를 통하여 물리적 프로세스의 Dynamics를 초정밀 실시간성으로 감지, 제어하는 시스템

Componentizing Kernel



- 각 소프트웨어 콤포넌트에 독립적인 하드웨어 콤포넌트를 할당함으로써 의존성 제거하여 콤포넌트간 상호 지연이 없게 하고 전체 시스템의 검증을 쉽게 함

Cyber-Physical Simulator



- 실제 하드웨어와 시뮬레이션 되는 하드웨어간 정확한 실시간 데이터 교환을 지원
- 시뮬레이션 되는 하드웨어가 실제 하드웨어로 바뀌는 개발 과정 전반을 유연하게 지원



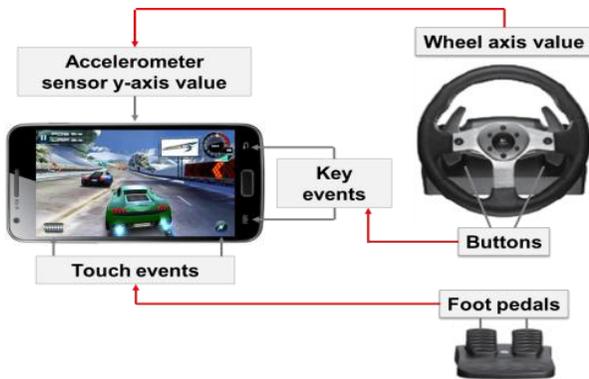
모바일 플랫폼 분야 연구



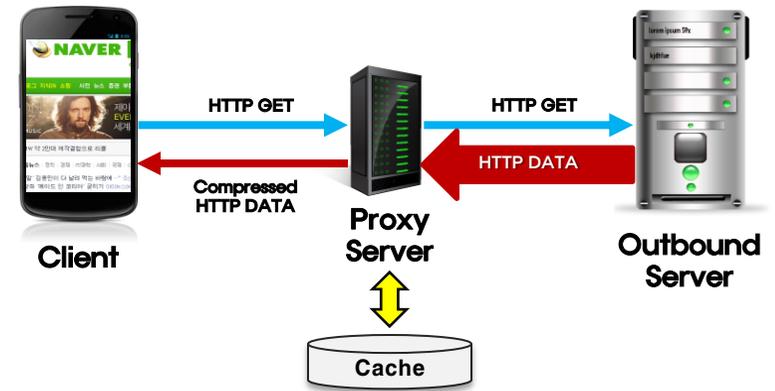
- 외부 디스플레이 장치의 해상도에 맞게 스마트폰 화면을 전환 (**Adaptive Display Switching Tech.**)
- 리모콘, 뷰파인더 등으로 멀티 스크린을 동시에 활용



- 모바일 어플리케이션의 입출력 모듈 뿐만 아니라 연산모듈을 외부 연산장치로 전환
- 스마트폰 성능, 배터리, UI 만족도를 동시에 고려한 최적 **Offloading** 기술 개발



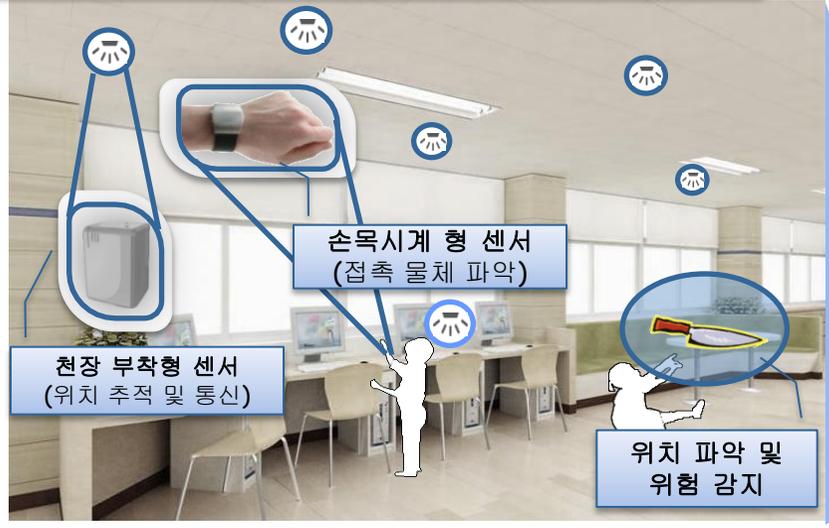
- 다양한 외부 입력장치를 활용하여 스마트폰을 제어 (**Adaptive Input Switching Tech.**)
- 레이싱휠, 조이스틱, 외부 센서 등을 기존 모바일 어플리케이션에서 그대로 활용



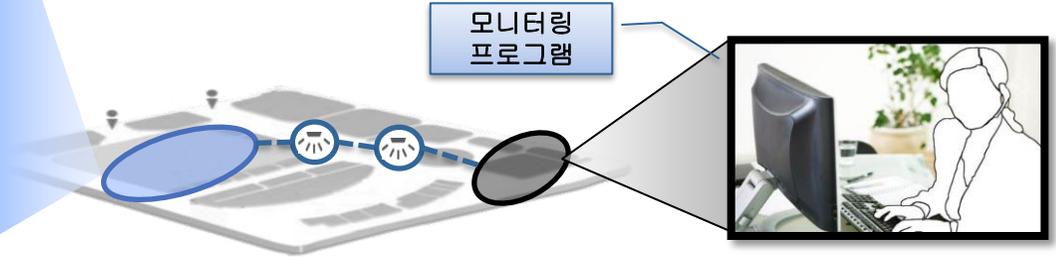
- **Proxy Server**를 이용해 모바일에 최적화된 웹 브라우저를 제공하는 **Proxy Offloading** 모바일 웹 브라우저개발

실내 위치 추적 기술 연구

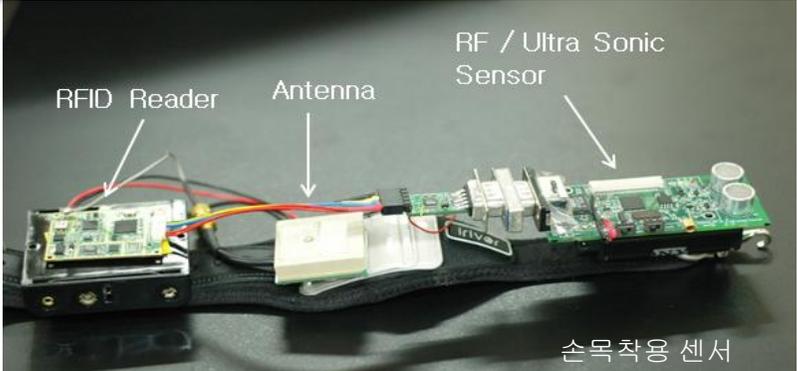
발달장애 아동의 위험상황 모니터링



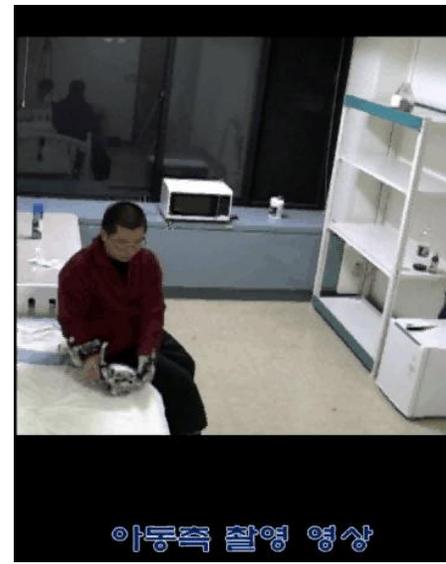
- 발달장애아동의 손목에 부착된 센서를 통해 언제, 어디서 무엇을 접촉하고 있는지 파악
 - 시간, 위치, 접촉 사물을 분석하여 위험상황 판단
- 위험상황 발생시 센서네트워크를 통하여 원격지에 있는 보호자에게 알림 발생



프로토 타입 구축



- 초음파와 RF 신호의 도착 시간차를 이용하여 거리 측정
- 측정된 거리로 부터 삼변측량법을 이용하여 위치 추정
 - RFID 리더를 통해 접촉 물체 파악



연구 주제

- Provably-correct Multi-threaded 플래시 메모리 관리 알고리즘
- 고성능 플래시 메모리 컨트롤러
- 고효율 오류 정정 알고리즘
- 플래시 메모리 Fault 모델링



Hynix MLC NAND

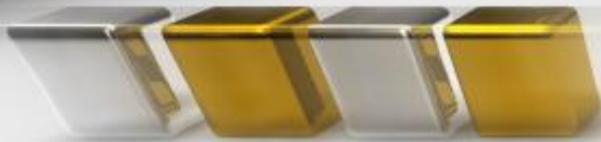


Samsung SLC NAND



Samsung MLC NAND





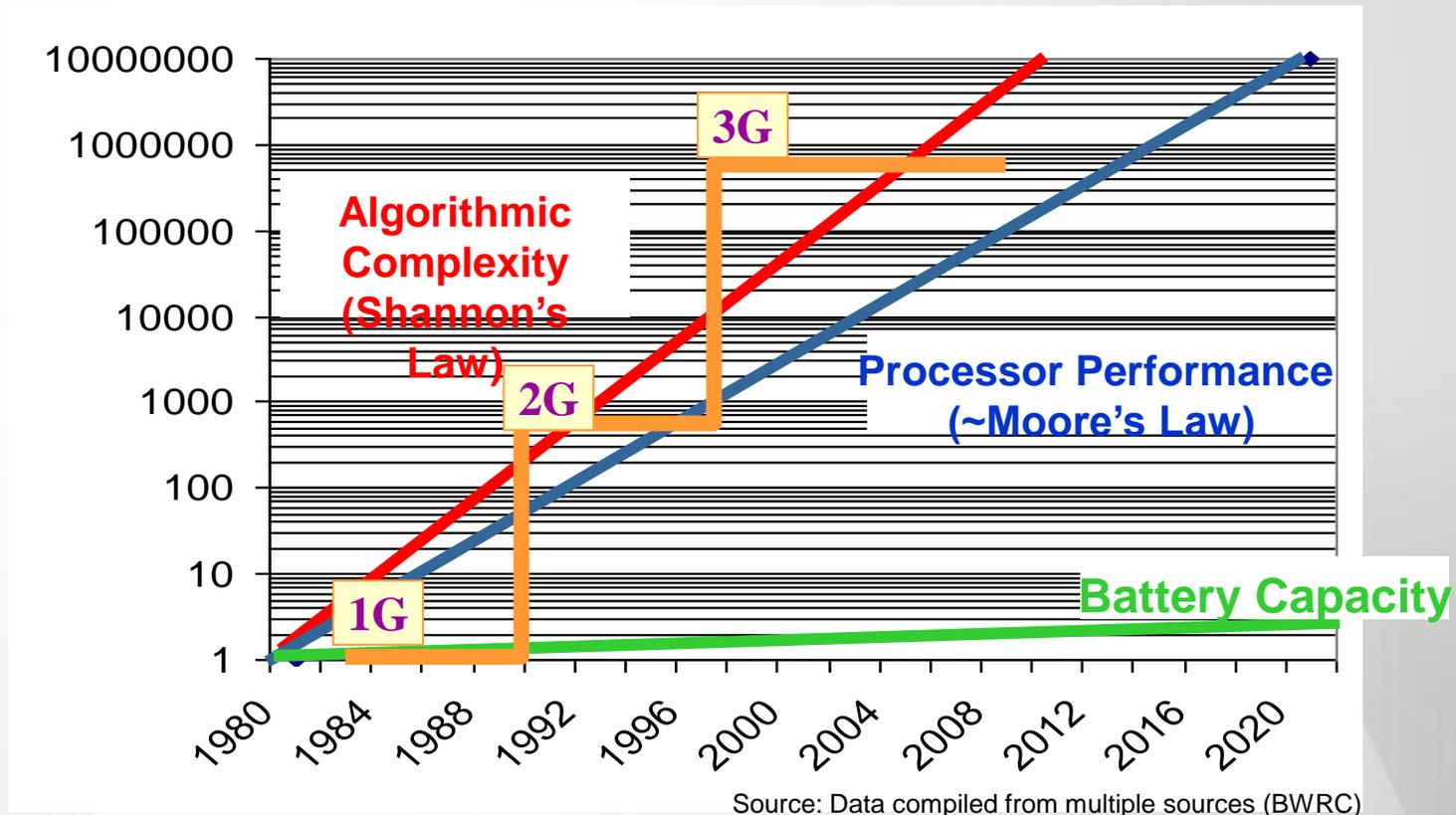
다시 본론으로 ...



시스템의 복잡도 증가

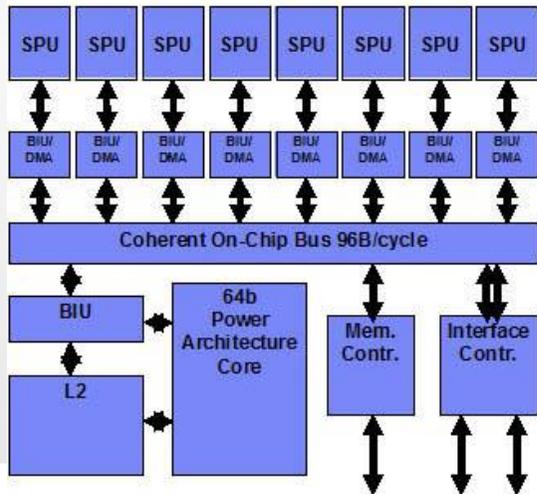
Complexity increases incessantly

- More processing power is required
- 저전력 기술이 매우 중요해짐

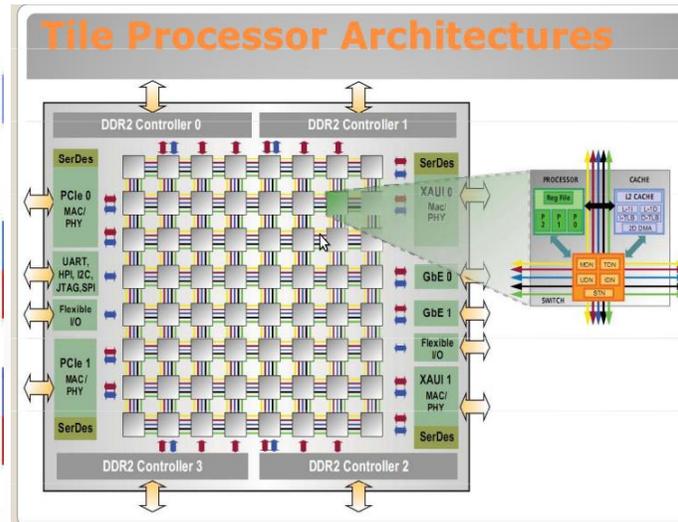


기술 현황과 전망

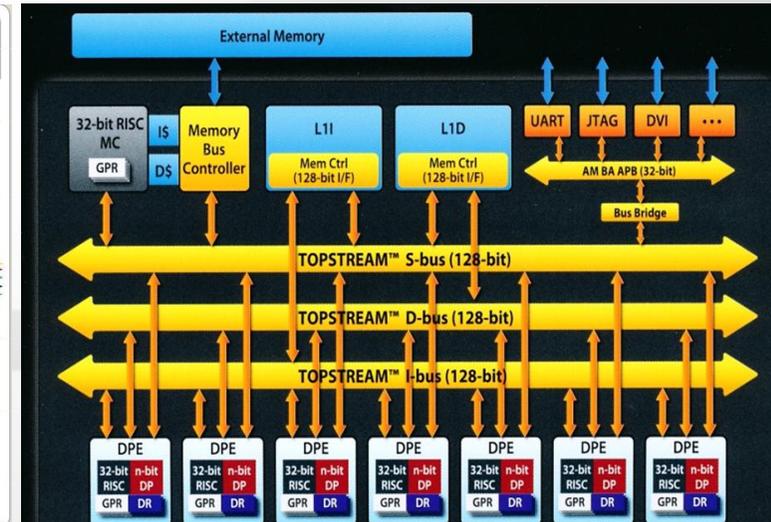
- 시스템에 집적되는 프로세서의 증가
 - Heterogeneous architectures
 - Many-cores: 수백, 수천개의 프로세서 시스템
- 다양한 응용이 동시에 수행됨 (dynamic behavior)
- Design Challenges
 - HW/SW co-development becomes more important
 - System verification becomes more difficult



IBM Cell



Tilera-64

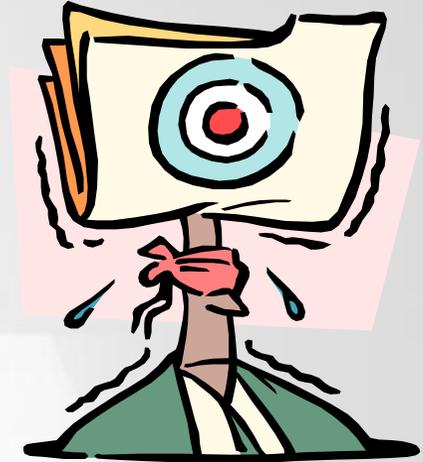


TOPSTREAM

How to Design an Embedded System

❖ Integration of diverse subjects into a whole

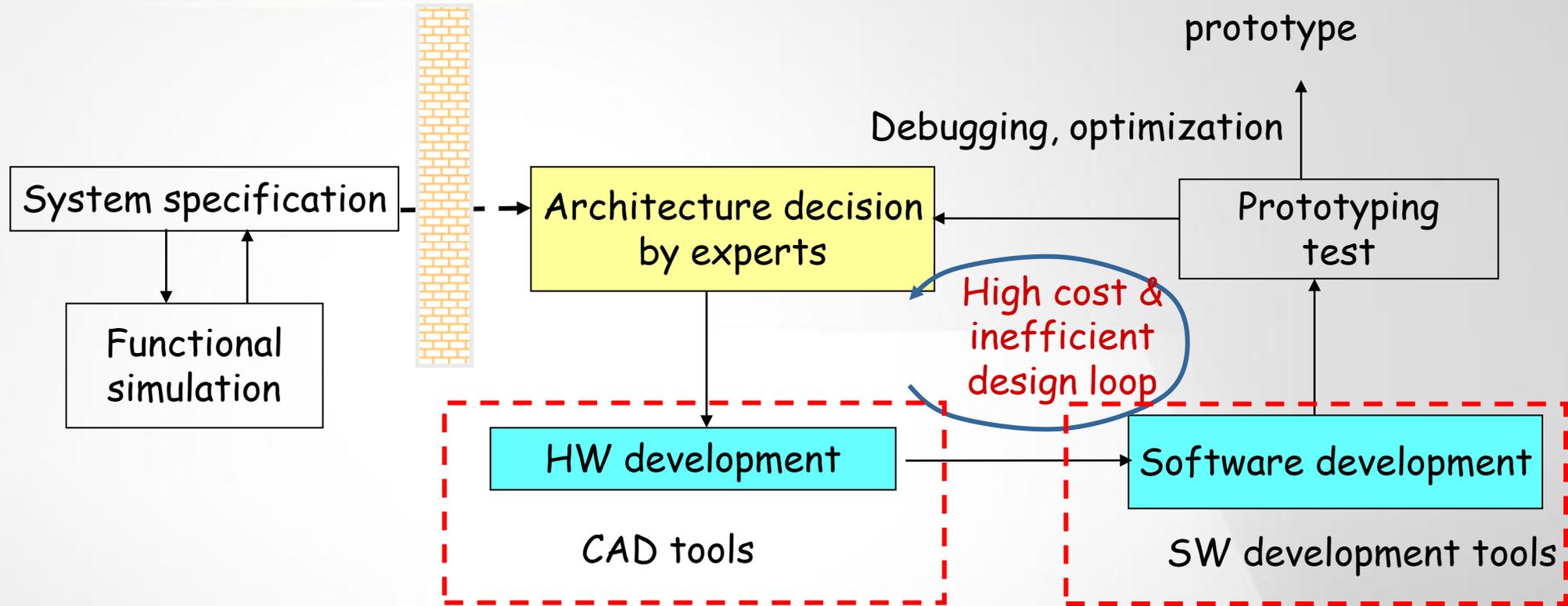
- Hardware architecture
- System software
- Application software
- Optimization algorithm
- Software/system engineering



❖ 복잡도 관리 기법

- **분해 (Decomposition)**: decompose a complex problem into small simple problems
- **추상화 (Abstraction)**: translate the problem into a simpler one by hiding unnecessary details

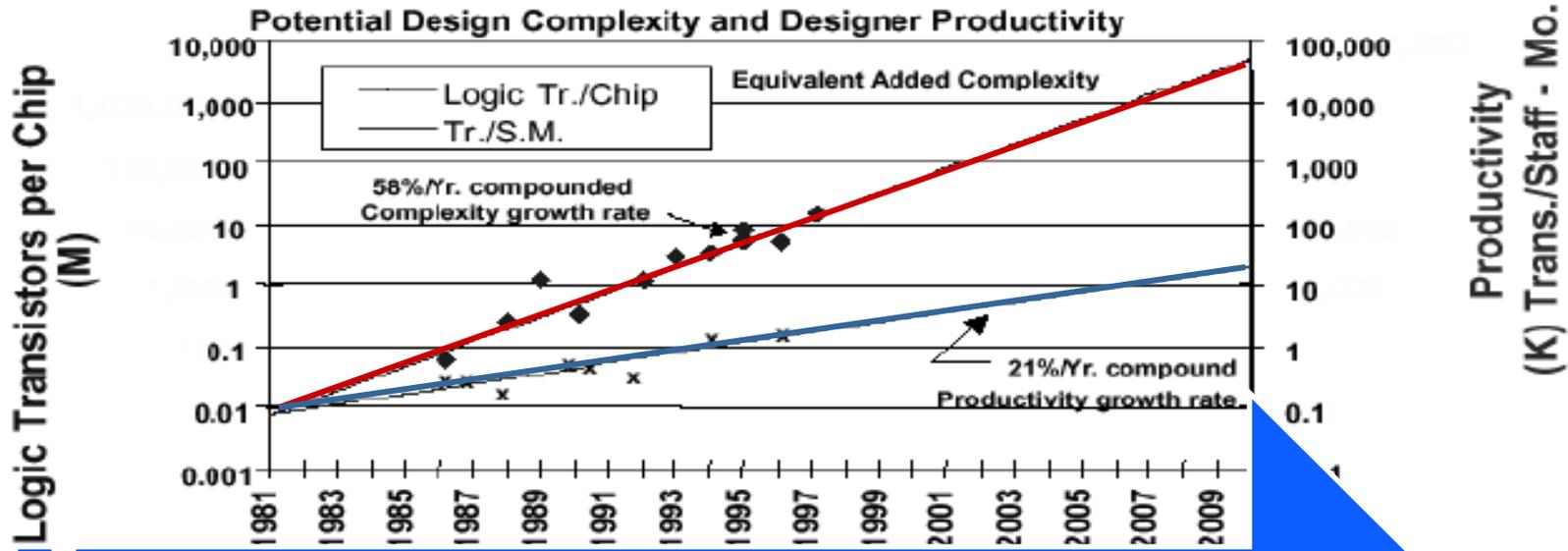
기존의 설계 방법



→ Early separation of HW and SW
Slow/ high cost/ hard to maintain and debug



생산성 간격 (Productivity Gap)

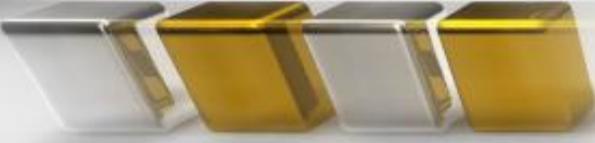


Need a New Methodology!

* @ \$150K / StaffYr. (In 1997 Dollars)

Source: SEMATECH





“새로운” 설계 기법

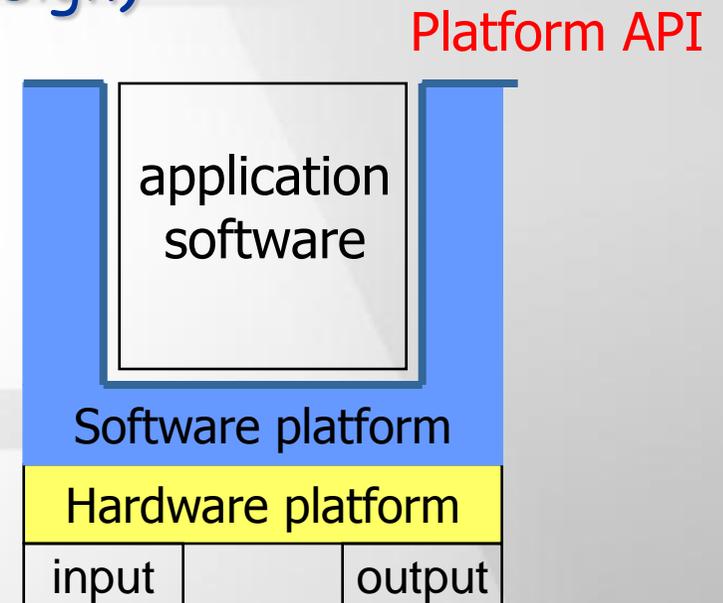
- ❏ (1) 설계재사용 (**Design Reuse**): Virtual Component (IP) based design & platform based design
- ❏ (2) **HW/SW** 통합설계 (**Codesign**): Systematic Design Methodology
- ❏ (3) 모델기반 설계 (**Model-based design**): formal model is good for verification
- ❏ (4) 시스템 수준 컴퓨터이용설계 도구 (**CAD**) 필요

IP-based design

- Reuse the pre-verified component
 - Processor core, peripheral, memory, interface
 - SW library

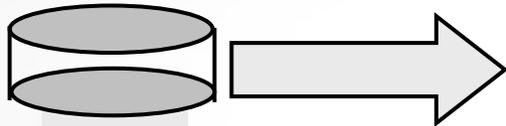
플랫폼 기반 설계 (Platform-based design)

- Reuse the pre-verified architecture
 - Trade-off between customized design and general purpose design

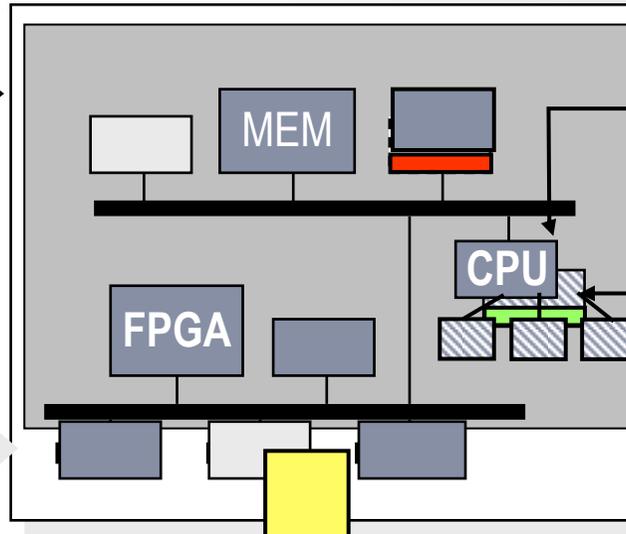


플랫폼 기반 설계

Pre-Qualified/Verified
Foundation-IP*



Foundation Block + Reference Design



Scaleable
bus, test, power, IO,
clock, timing architectures

Processor(s), RTOS(es) and
SW architecture

Methodology / Flows:
System-level performance
evaluation environment

Rapid Prototype for
End-Customer Evaluation
SoC Derivative Design
Methodologies

Foundry Targetting Flow

*IP can be hardware (digital or analogue) or software. IP can be hard, soft or 'firm' (HW), source or object (SW)

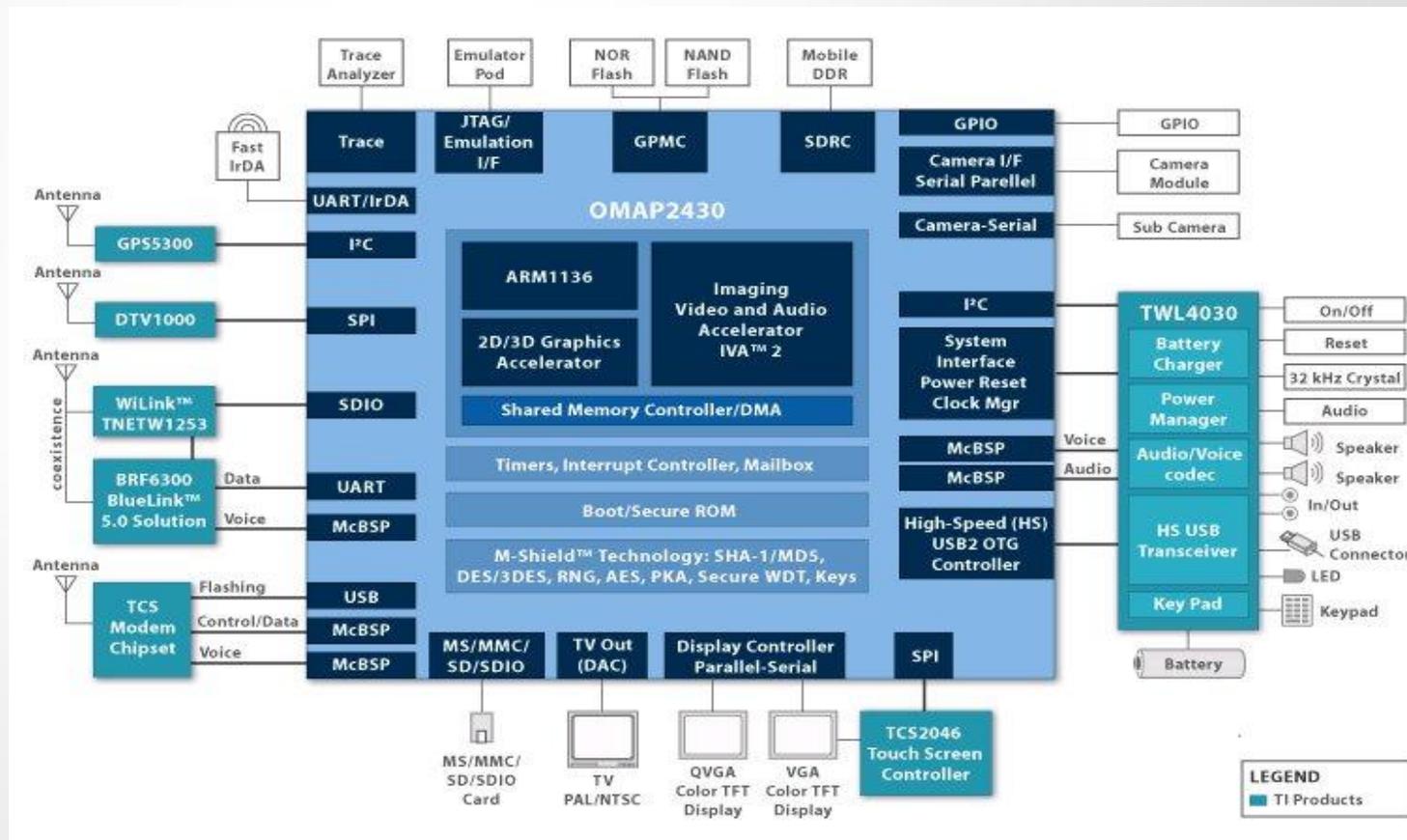
Foundry-Specific
Pre-Qualification

Source@cadence



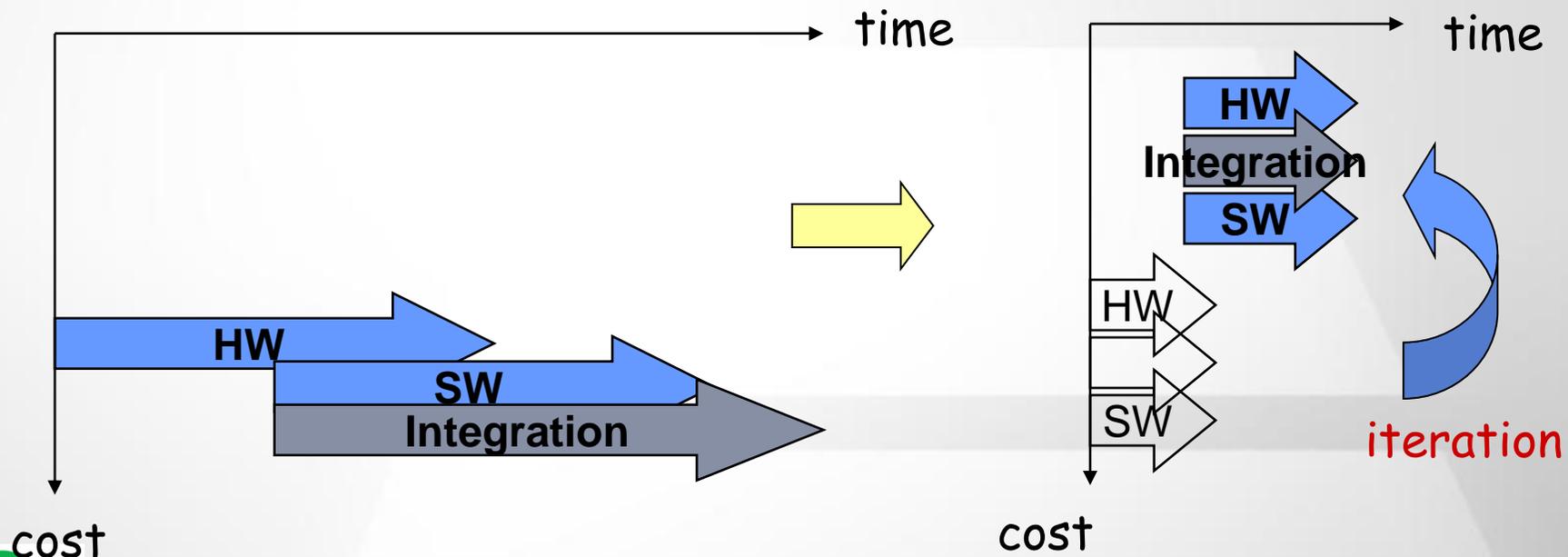
플랫폼의 예: TI OMAP

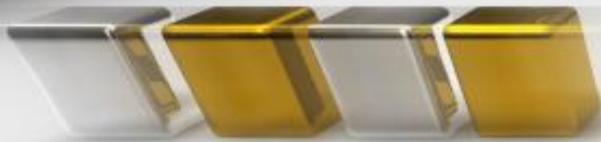
- ARM-based platform
- Accelerators for Graphics and Video/Audio CODECs



HW/SW 통합설계

- Concurrent design of HW/SW components
- Evaluate the effect of a design decision at early stage by “가상프로토타입(virtual prototyping)” to enable systematic 설계공간탐색(design space exploration)

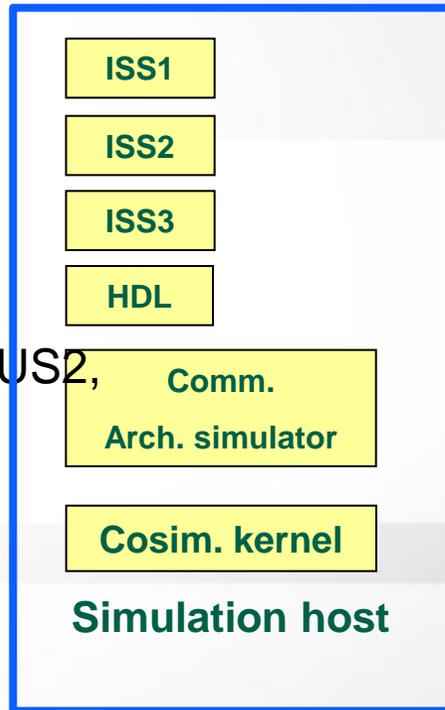




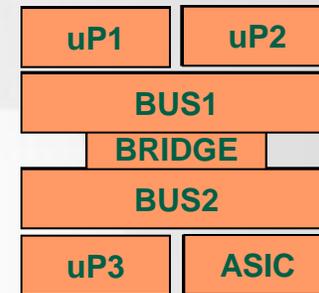
HW/SW 통합 시뮬레이션

- 가상 프로토타이핑: 컴퓨터 시뮬레이션으로 실제 하드웨어 플랫폼을 모의 실험

Virtual Prototype of a System

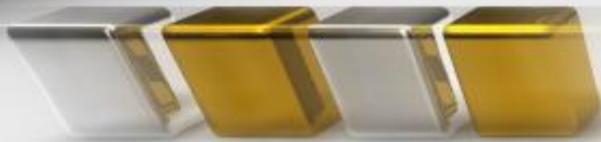


Real Prototype of a System

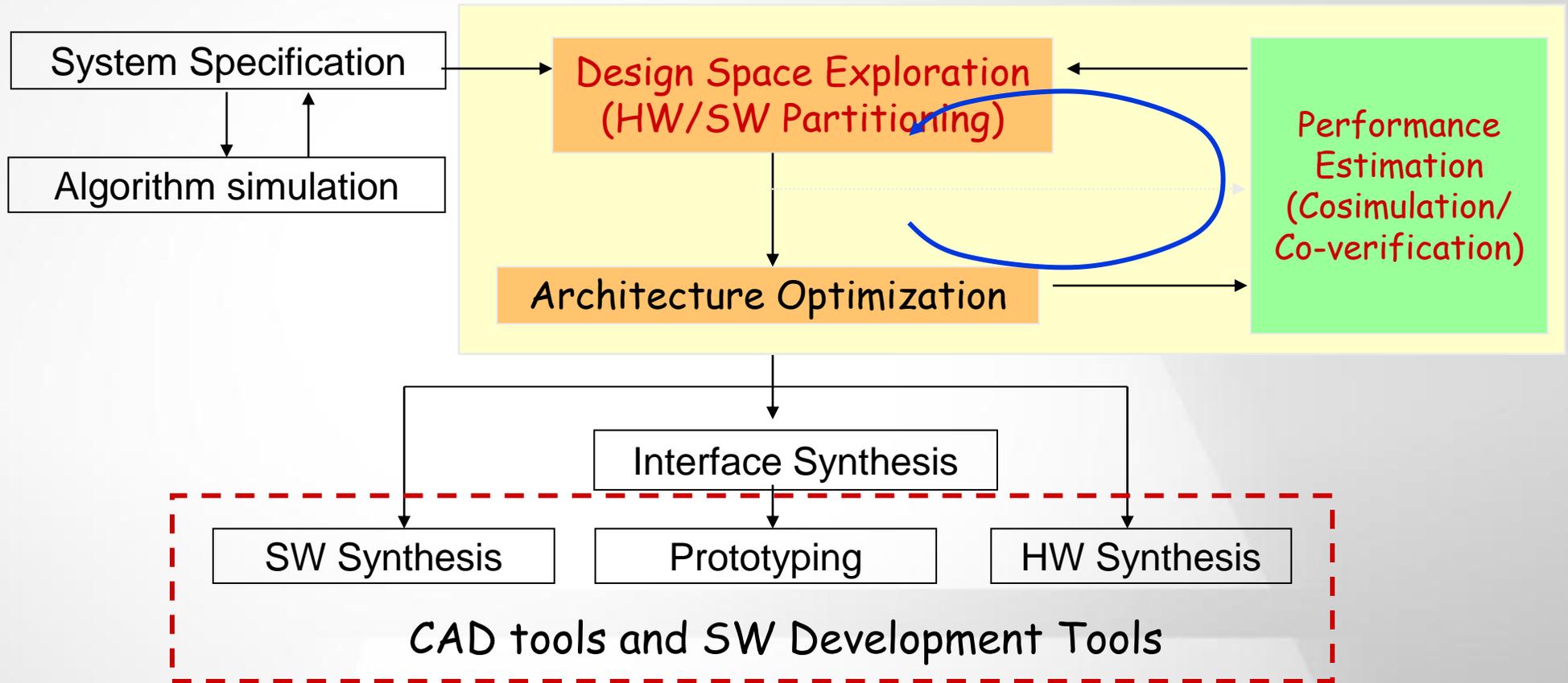


component simulator for uP1
component simulator for uP2
component simulator for uP3
component simulator for ASIC

comm. arch. simulator for BUS1, BUS2,
BRIDGE



HW/SW 통합 설계 흐름



좋은 모델의 조건

- Implementation (hardware or software) independent
- **검증이 용이해야 함**
- **Refinement** from functional simulation to implementation
- Modeling languages
 - HW: HDL (Hardware Description Languages: VHDL, Verilog)
 - SW: C, C++, Java
 - **시스템 수준 모델링 언어**: SystemC, SpecC, Esterel, Dataflow

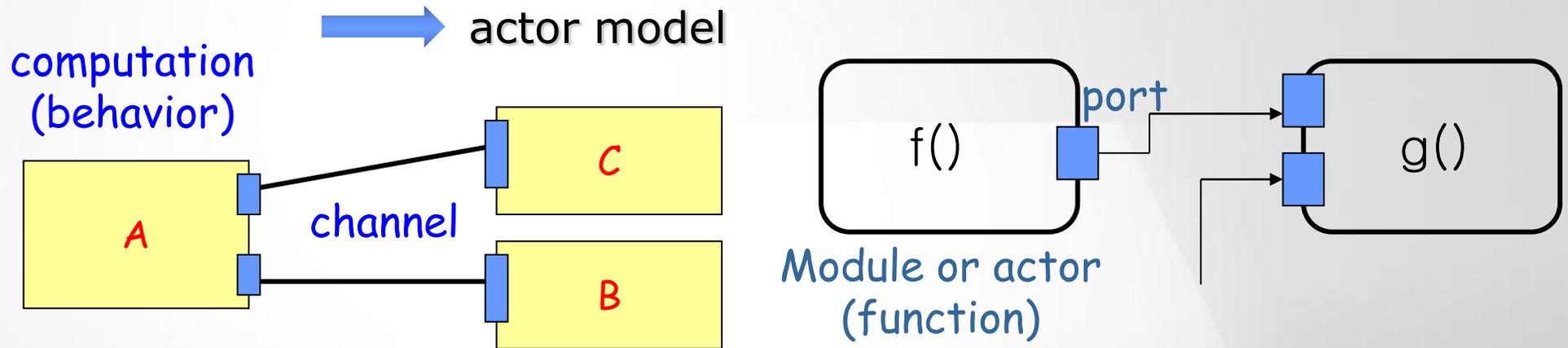
프로그램으로부터 시스템을 설계 하는 기법: 기존 방법

- Algorithm specification in a programming language, C or C++.
- Good for functional simulation
- **병렬 시스템 설계에 적합하지 않음**
- **성능 분석이 어려움**



☛ 모델을 이용한 응용의 표현

- 병렬성이 잘 표현되어야 함
- 계산 부분과 통신 부분의 분리가 필요함

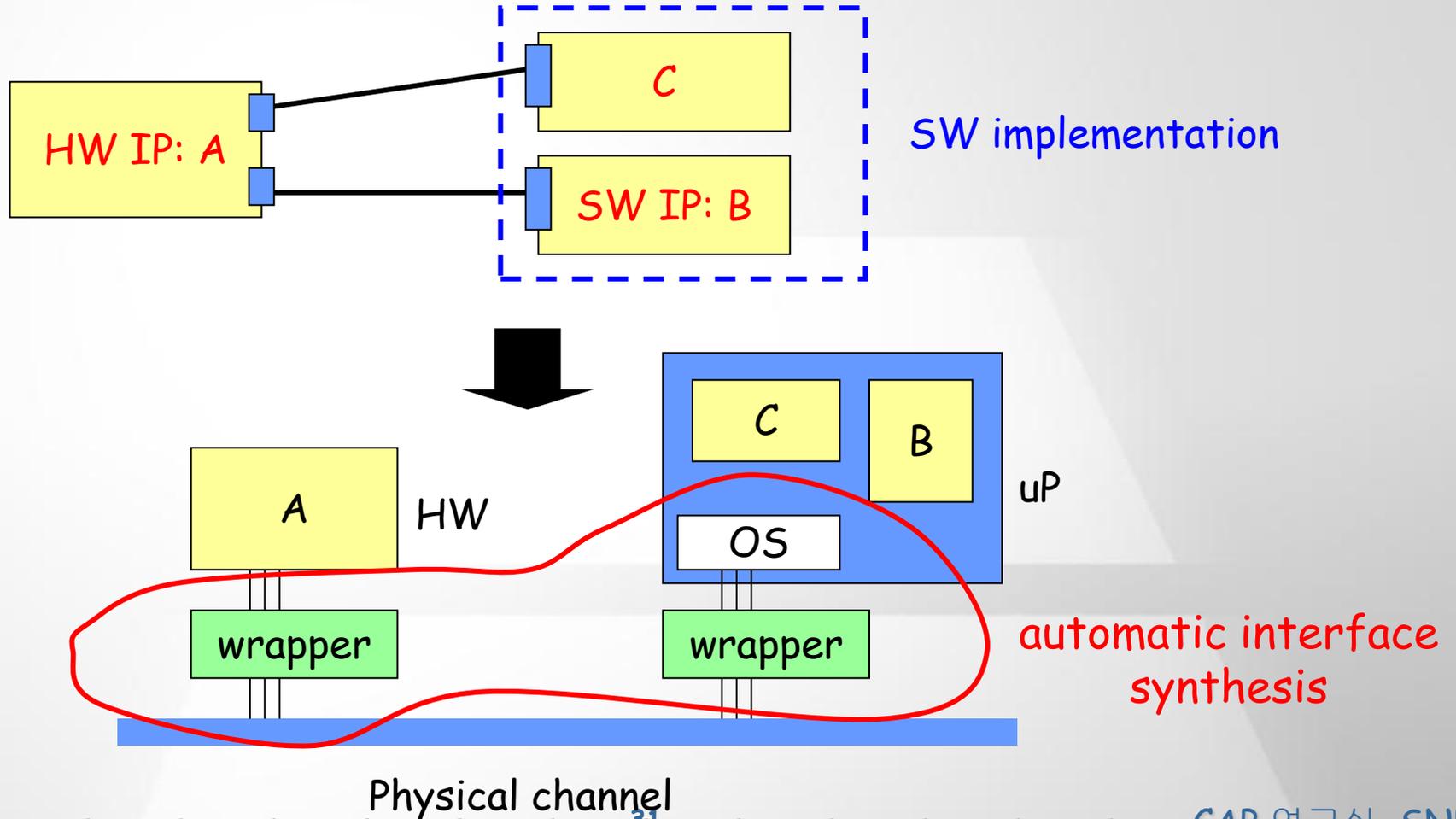


☛ 정형적인 모델: precise and unambiguous semantics

- Functional specification: $f(\text{input, output, state})$
- Well defined function composition
- Properties and Constraints

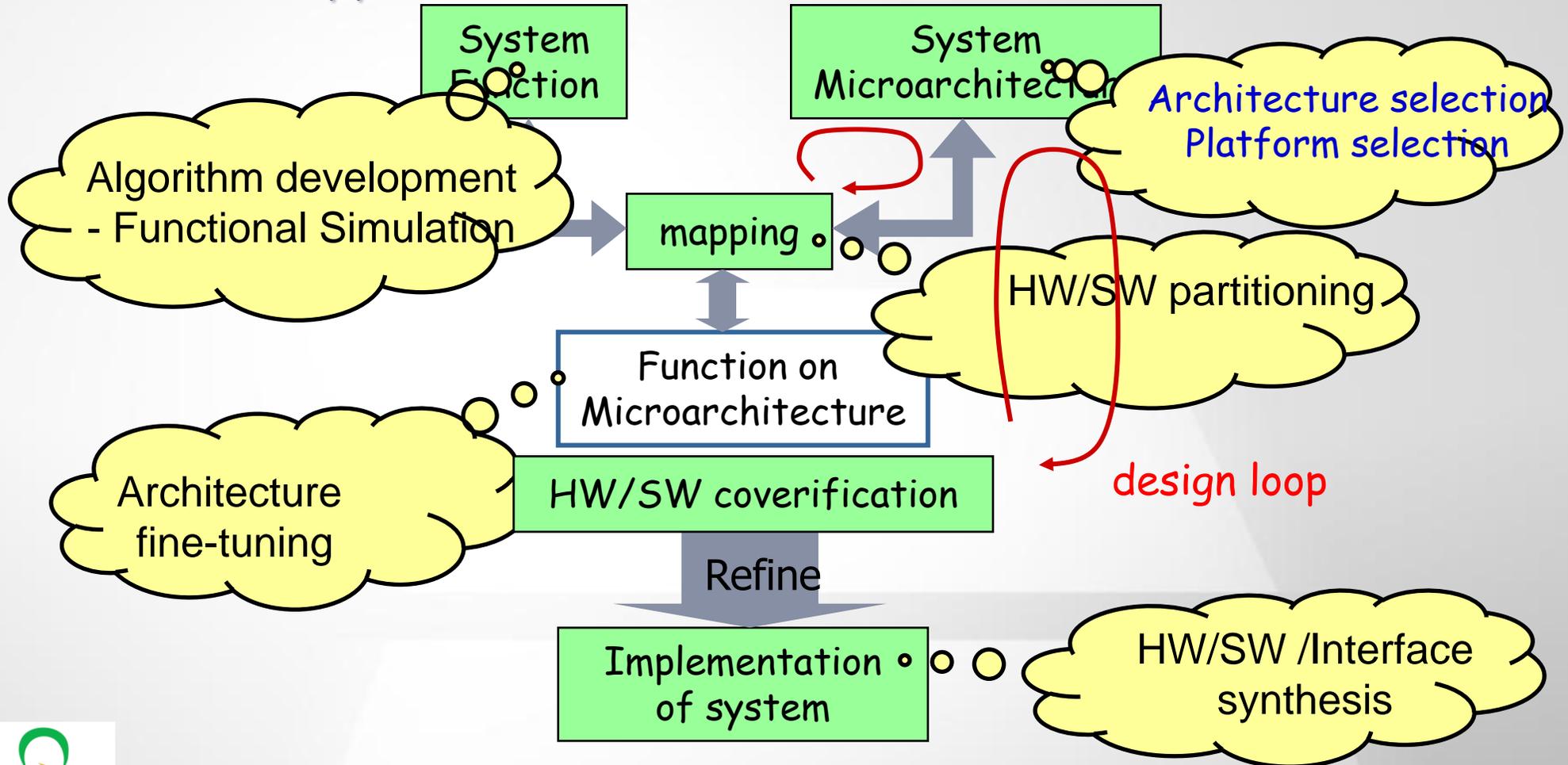
모델 기반 설계의 장점

- 다양한 방법으로 시스템을 구현할 수 있다
- Synthesis is refinement



매핑에 의한 설계 공간 탐색

Y-chart approach



📦 PeaCE home page

- <http://peace.snu.ac.kr/research/peace>
- papers, manual, etc

📦 Open source program

- Released on Nov. 2004.
- Demonstrated at the Univ. Booth in DAC 2003-2006

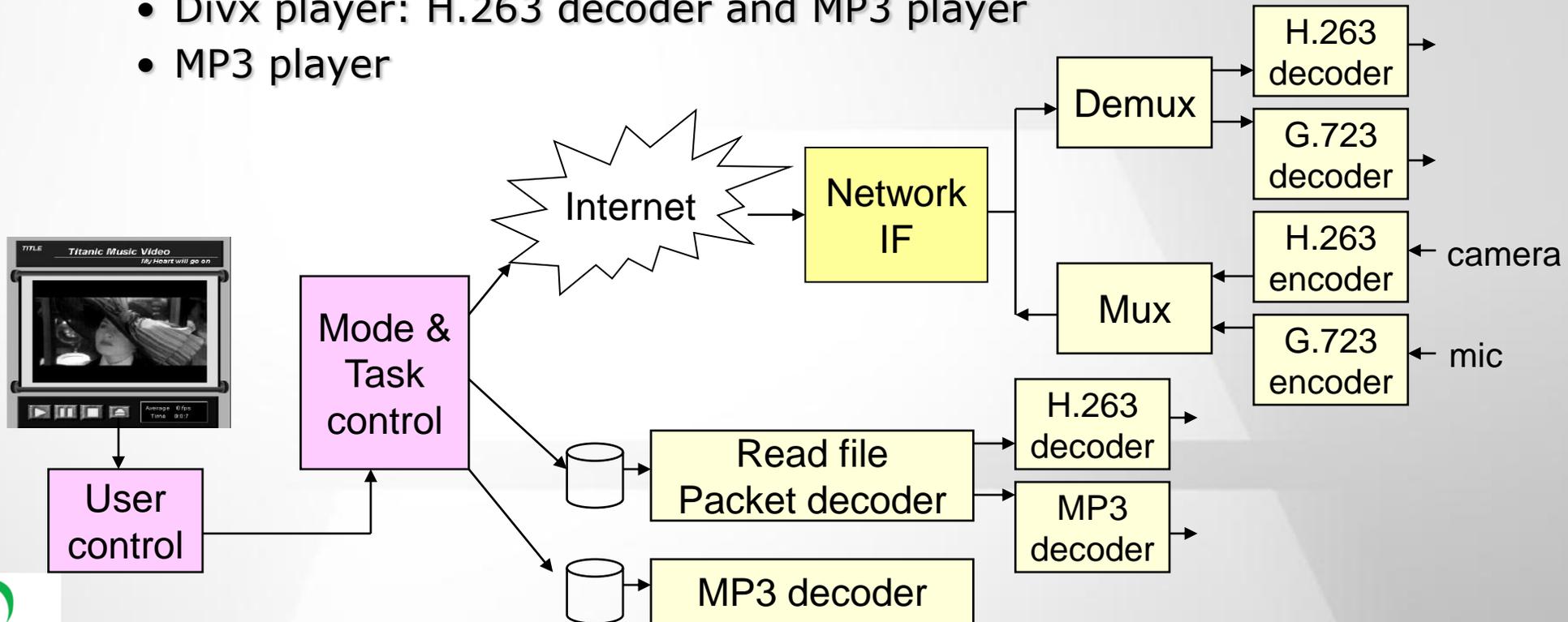
📦 Provide all design steps from system-level specification to synthesis



Motivational Example

Multi-mode Multi-media Terminal Example

- Three modes of operation:
 - Video phone: H.263 encoder/decoder and G.723 encoder/decoder
 - Divx player: H.263 decoder and MP3 player
 - MP3 player



PeaCE Design Environment

PeaCE : Ptolemy extension as Codesign Environment

File Edit View Run Tool Library Option Comment Help

Design Lib Platform

Design

- MMT_API.basicProject1230
 - MMTFSM_Top14
 - VIC_USB_API2

CGC Task-Model

Name	Value	Type
processor	arm720T	STRING
processorid	0	INT
host		STRING
directory	\$HOME/PEA...	STRING
file		STRING
Looping Level	SJS	STRING
display?	NO	INT
compile?	YES	INT
run?	YES	INT
write schedul...	NO	INT
staticBuffering	YES	INT
funcName	main	STRING
compileCom...	gcc	STRING
compileOpto...		STRING
linkOptions	-lm	STRING
resources	STUDIO	STRINGARR...
optLevel	0	INT
bufferSharing	NO	INT
numUsable	1	INT
seq	1	INT
nonseq	1	INT
DebugMode	0	INT

Add Parameter Del Parameter

Name	Value	Type
width	176	int
height	144	int
MBCols	176/16	int
MBRows	144/16	int
enable	1	int

design tabs

Task Model

FFSM Model

Extended SDF (dataflow) Model

The screenshot displays the PeaCE Design Environment interface. On the left, there is a 'Design' tree showing a project named 'MMT_API.basicProject1230' with sub-projects 'MMTFSM_Top14' and 'VIC_USB_API2'. Below the tree is a 'Task-Model' configuration table with various parameters like 'processor', 'host', and 'resources'. The main workspace contains three design tabs: 'sim', 'arch', and 'estimation'. The 'sim' tab is active, showing a 'Task Model' diagram with components like 'VIC_USB_API', 'mp3_test_API', 'UIGal_new_API', and 'MMTFSM'. The 'arch' tab shows an 'FFSM Model' (Finite State Machine) diagram with states like 'divvx', 'self', 'selfstop', 'mp3', and 'STAR'. The 'estimation' tab shows an 'Extended SDF (dataflow) Model' diagram for an 'H.263 Encoder in fractional rate', including components like 'H263Enc', 'H263Dec', and 'H263EncStream'. The bottom of the window has a status bar with 'log' and 'run' buttons.



PeaCE Demo : Functional Simulation과 프로토타이핑

./schematic/User/mykim/MMT_test

VIC_USB DivX_task_CGC mp3_test

connectGal

MMT Terminal

start suspend stop dir

Select

mp3://kings.mp3
divx://friend.avi
phone://147.46.121.1

H,263 Display

CANCEL OK





HOPES 프로젝트

📦 Target: 매니코어 이종 임베디드 시스템

- 확장성을 고려함
- 시간과 전력 제약 조건을 가짐

📦 해결할 과제들 (Sub-problems)

- Parallelism extraction (multiple use case, multi-tasking apps.)
 - Functional parallelism, data-parallelism, temporal-parallelism
- Partitioning and mapping
- Parallel code generation: parallel programming is not easy
- Performance estimation and verification
- Design space exploration

➡ Need of sound (scalable and robust) methodology



병렬 임베디드 SW 개발 기법들

❏ 컴파일러 기반 방법

- MAPS from RWTH, Aachen

❏ 언어 확장 기법

- Language extension with annotations
 - openMP for IBM Cell
- Language extension with APIs
 - CUDA from nVidia, OpenCL from Khronos group, Ct from Intel, Stream tools for SPI, Hierarchical Tiled Array from UIUC

❏ 모델 기반 방법

- Gedae, StreamIt from MIT, Sesame from U. Amsterdam, Simulink

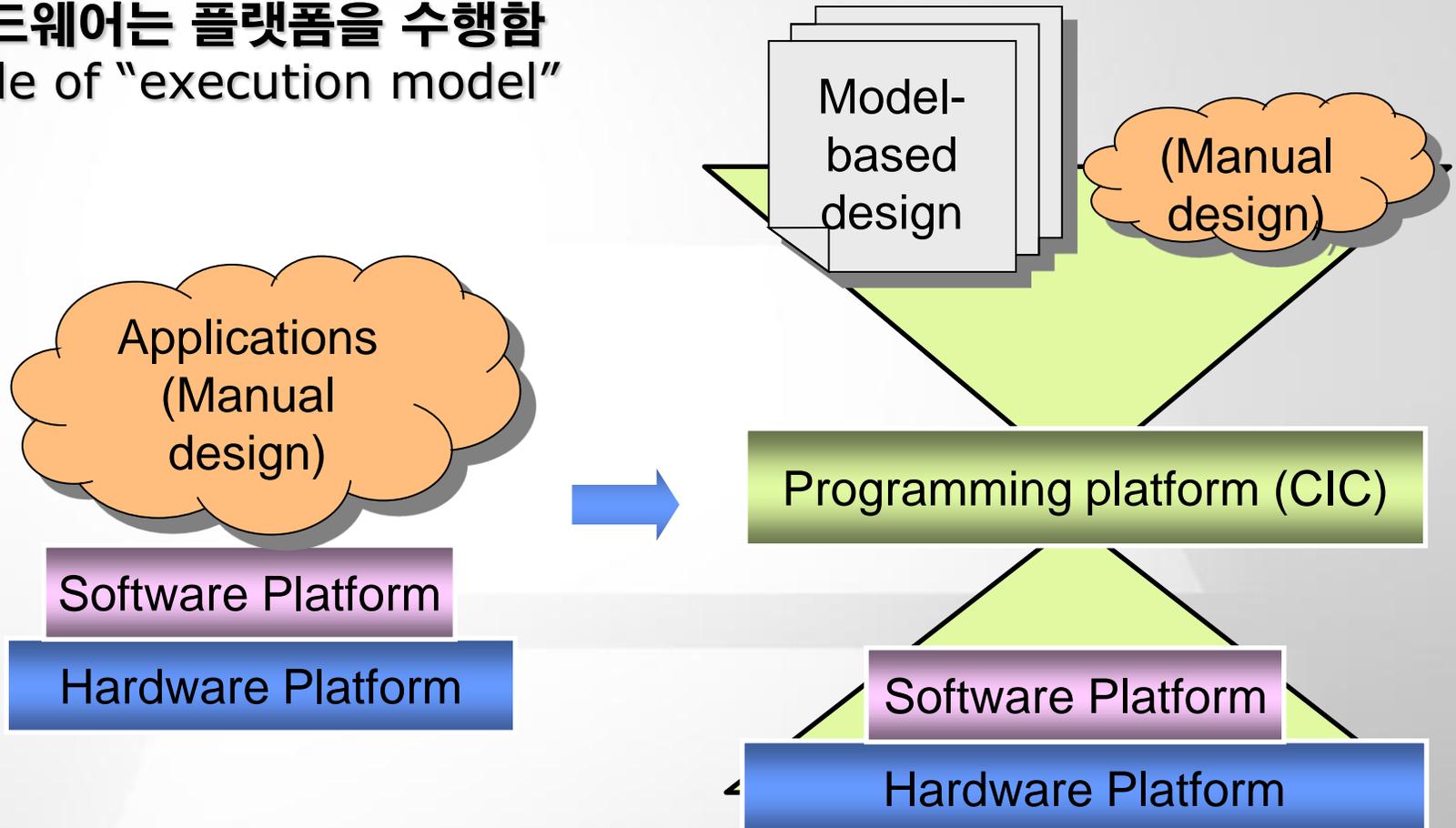
❏ 플랫폼 기반 방법: **HOPES**



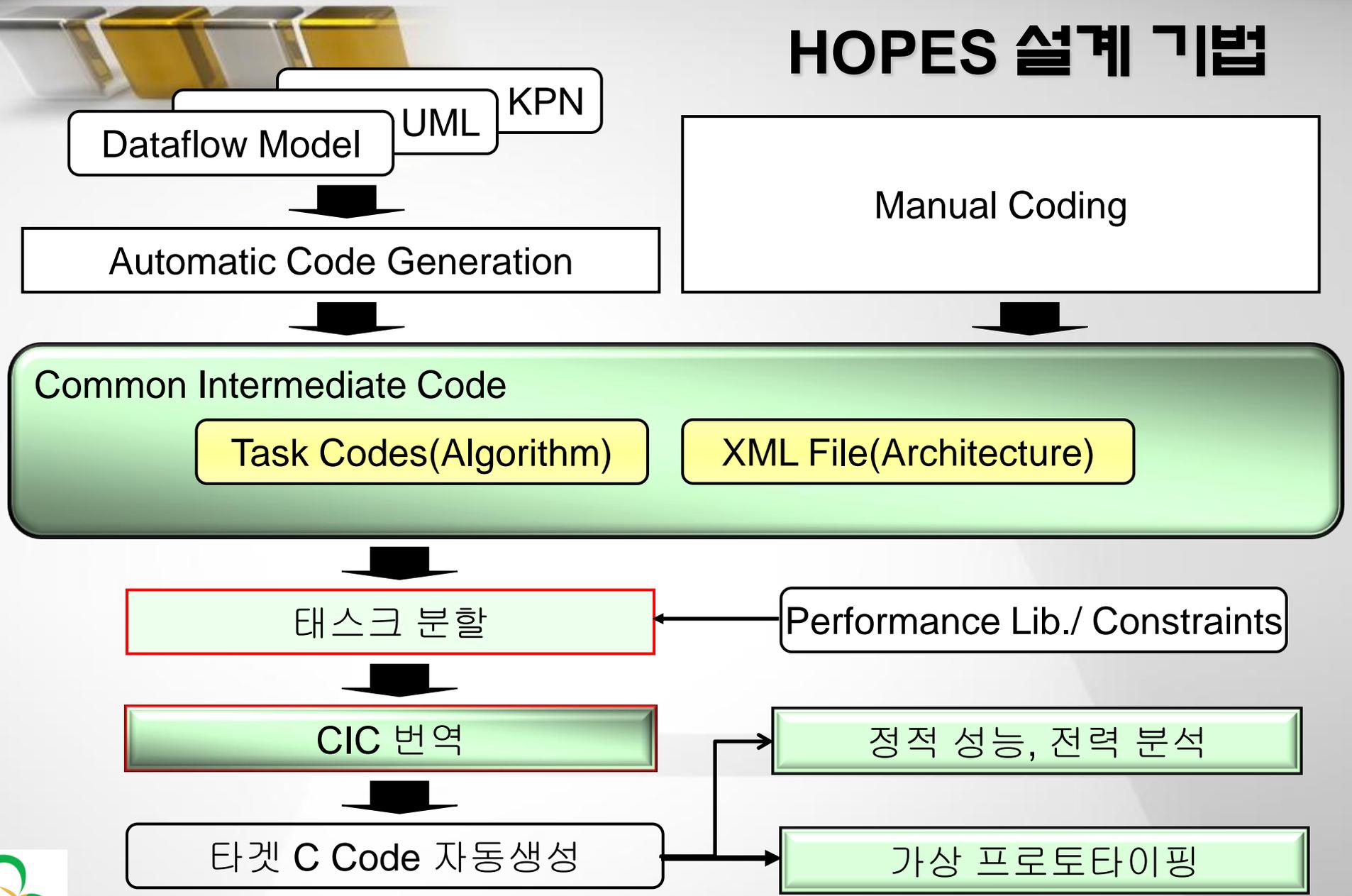
새로운 아이디어: 병렬 프로그래밍 플랫폼!

❖ 프로그래밍 플랫폼: meet-in-the-middle approach

- 프로그래머는 하드웨어를 고려하지 않음
- 하드웨어는 플랫폼을 수행함
Role of "execution model"



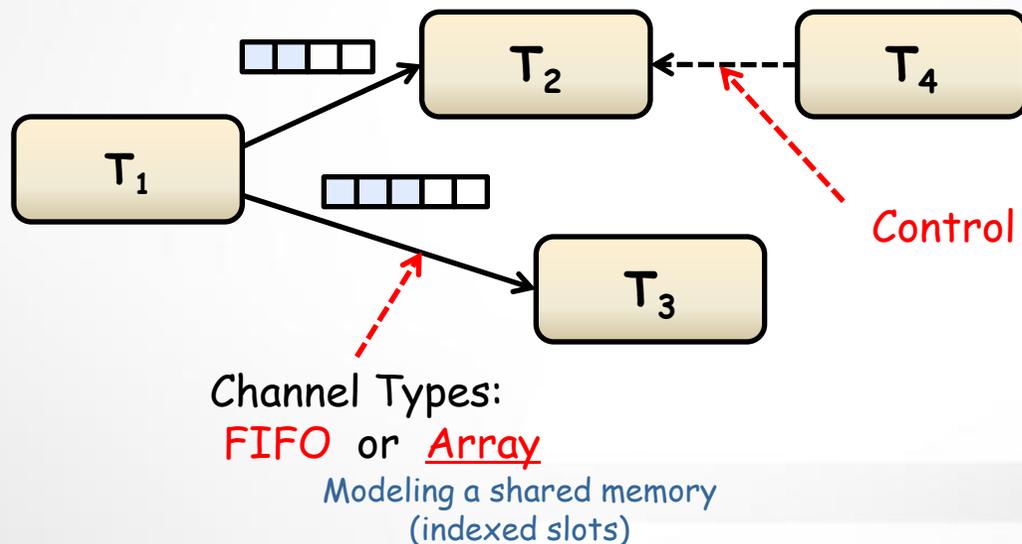
HOPES 설계 기법



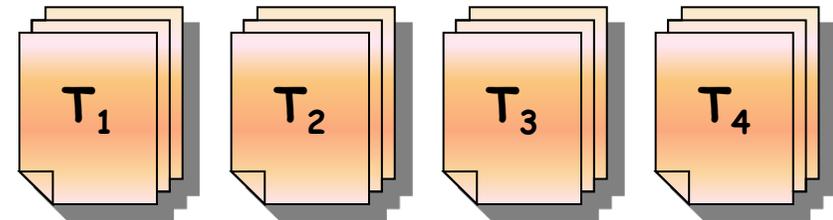
CIC (Common Intermediate Code)

❏ 프로그래밍 플랫폼

- execution model of a **parallel architecture**
- **일종의 모델 기반 방법**: Viewed as a “generalized dataflow model” or a process network



▪ CIC Task Codes



- **Algorithm**
- Avail. Parallelism
- **Architecture**
- **Mapping**
- **Control**
- **Profile**



❏ CIC 코드의 분석

- Functional simulation **discrete-time simulation w/ control tasks**
- **Static analysis**
- Parallel scheduling

❏ 지금까지의 타겟 프로세서 **Targets:**

- HSIM, IBM Cell, ARM MPCore, NXT+PC, **GPGPU(CUDA)**

❏ 가상 프로토타이핑

- Scalable, parallel simulation

❏ 최적의 아키텍처

For functional validation

- CIC translation to multi-thread codes (pthreads).
- Generated codes are run on a host machine

For performance estimation

- CIC 태스크를 타겟 프로세서로 매핑
- 매핑 결과를 이용하여 최악시간가 자원 요구량 예측

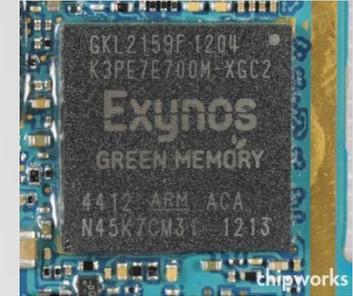
For performance verification

- CIC translation to target C code.
- 가상 프로토타이핑 시스템을 이용하여 검증

최근 Projects

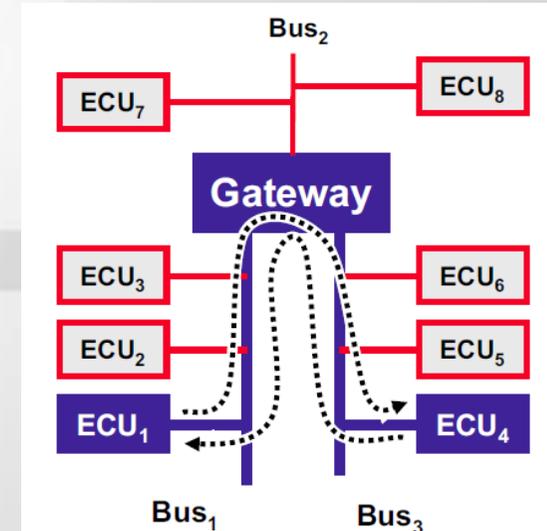
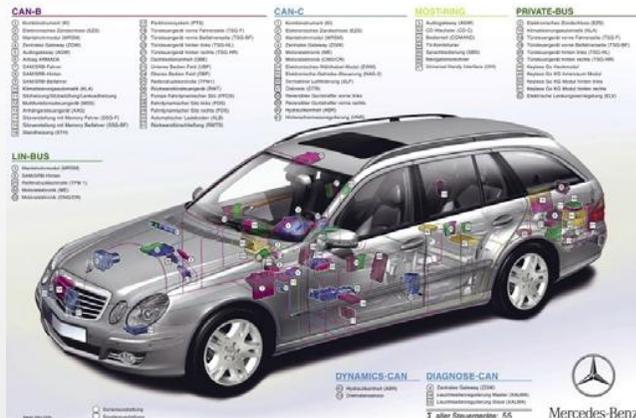
삼성 프로젝트

- 차세대 스마트폰의 가상 프로토타이핑 기술



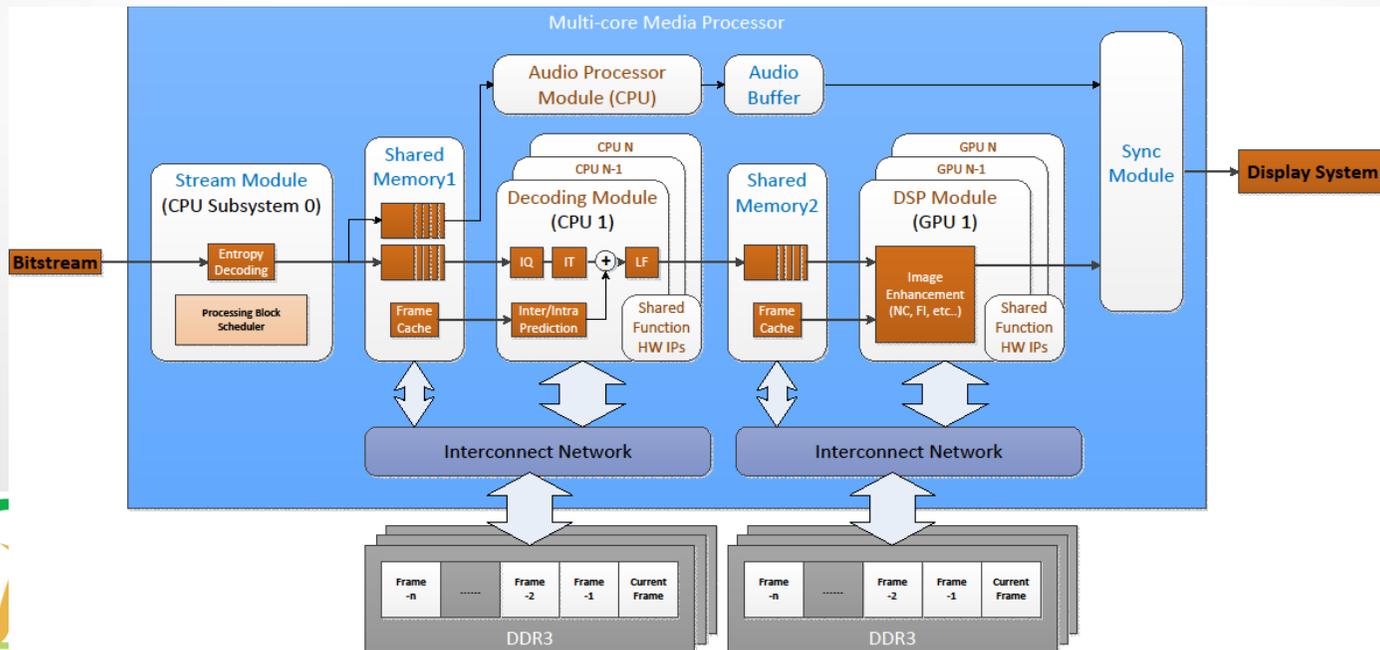
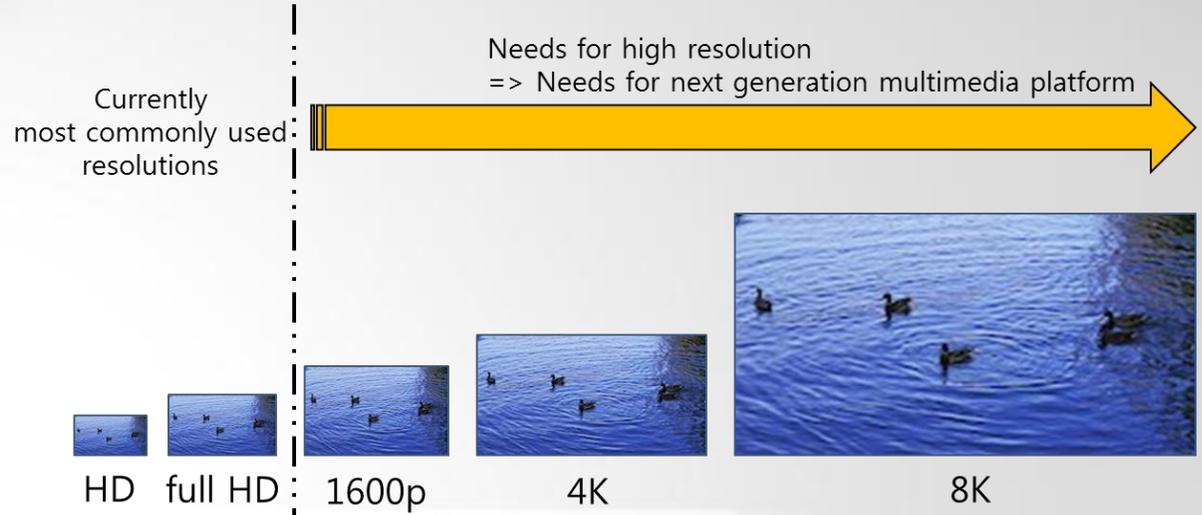
현대자동차 산학 프로젝트

- 센서부터 액추에이터까지의 종단간 최악시간 정적 분석



ITRC 위탁과제

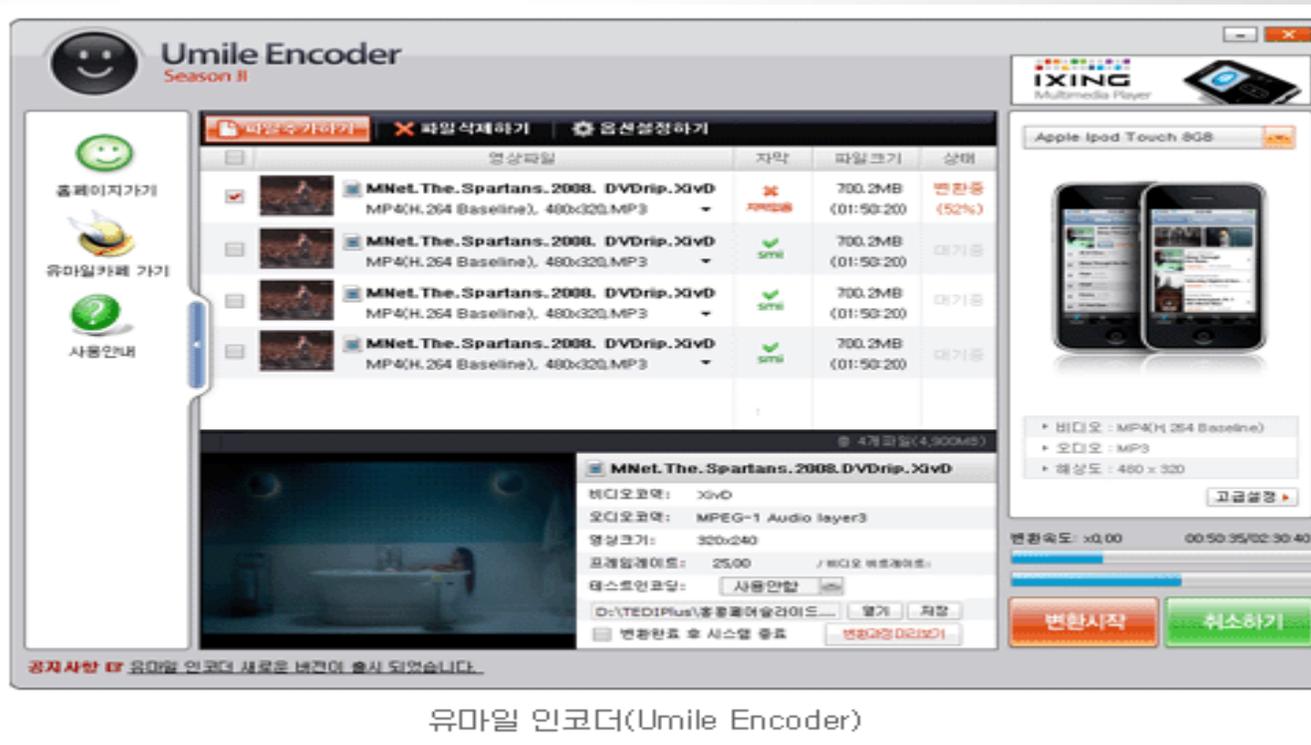
- 차세대 TV (4k) 플랫폼 구축



최근 Projects

☛ 서울시산학연 과제: Technonia(주) Umile 프로젝트

- 2009.12.1 - 2011.11.30
- GPGPU 를 이용한 비디오 트랜스코딩의 병렬화 기술 개발

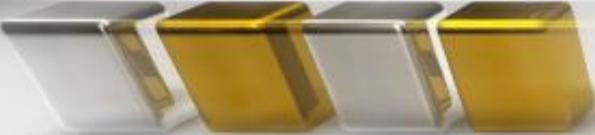


유마일 인코더(Umile Encoder)



- 임베디드 시스템 설계는 컴퓨터 공학/과학 여러 분야를 망라하는 복합 기술이다: 버즈 두바이를 만드는 기술보다 쉽지 않다!
- 사회 전반에 끼치는 영향력이 매우 크다.
- 기술과 인력에 대한 사회적인 수요가 매우 크다.
- 기술 선진국과의 격차가 적다.
- 새로운 설계 기법
 - 설계 재사용
 - HW/SW 통합 설계
 - 모델 기반 설계
 - 시스템 수준 CAD 필요





감사합니다!

