A case study of mobile SoC architecture design based on transaction-level modeling

Eui-Young Chung

School of Electrical & Electronic Eng.
Yonsei University

Outline

- Introduction
- Baseline architecture of Mobile-AP
- Design requirement
- Design strategy
- Architecture design methodology
- Architecture exploration
- Summary

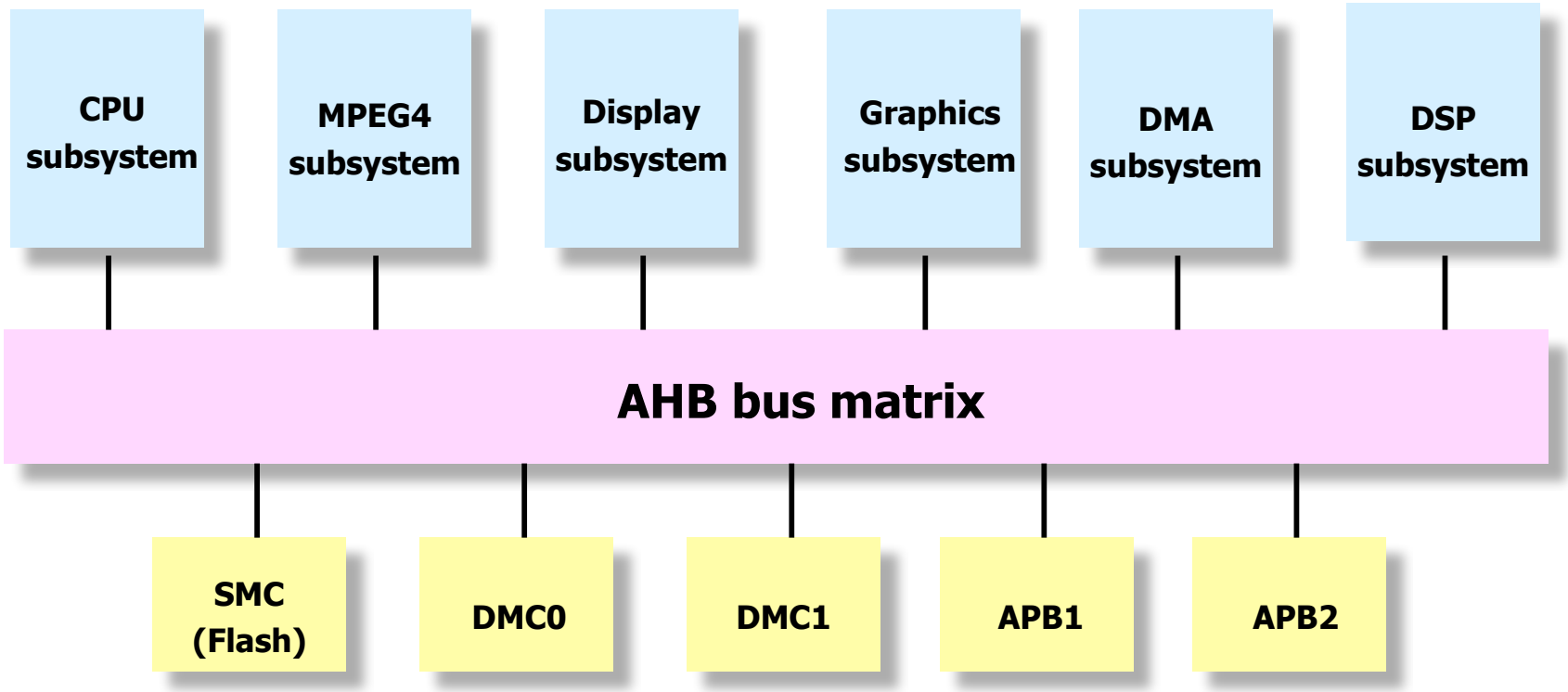
Mobile digital convergence

- **Smart mobile phone**
  - Integration of variety functions
  - Center of ubiquitous media network
  - Driving semiconductor industry
- **Internal components of mobile devices**
  - Display Driver IC (DDI)
  - CMOS Image Sensor (CIS)
  - Application Processor (AP)
    - Controls overall system
    - A major computing unit especially for multimedia applications
  - Connectivity: WLAN, Bluetooth
  - Modem: CDMA, GSM/GPRS, OFDM
  - RF/Analog
  - Volatile memory: DDR, SDR

Major features of mobile AP

- **High performance**
  - **Supporting VGA H.264**
  - **3D graphics**
  - **Multi-function → concurrent task execution**
- **Low power**
  - **130nm → 90nm**
    - **Leakage power management**
- **Small form factor**
  - **Smaller and smaller**
- **OS flexibility**
  - **Supporting various OS depending on the product lines**

Baseline architecture of AP



- Starting from the architecture of previous design
- Each subsystem is based on AHB
- DSP processor for audio and still image processing
  - Higher power consumption

Example of performance specification

Items	Old	New
Display size	CIF	VGA
Display colors	High color	True color
MPEG4 decoding	CIF, 30fps	VGA, 30fps
MPEG4 encoding	CIF, 30fps	VGA, 30fps
H.264 decoding	None	VGA, 30fps
H.264 encoding	None	VGA, 30fps
Camera I/F	1M pixel/sec	4M pixel/sec
3D graphics	.6M triangles/sec	1M triangles/sec

Other requirements (I)

- **Pin count**
  - Limits # of external memories

  → Smart memory controller
- **Power consumption**
  - Use MTCMOS technology for static power reduction
  - Retention F/F is used → Area overhead

  → Small area overhead of new architecture
- **Picture size**
  - Nice to have larger size of picture for H.264 as well as MPEG4

  → High performance bus architecture

Other requirements (II)

- **Clock speed**
  - **133MHz, but higher the better**
  - **➔ Bus-level pipelining (register slicing)**
  - **➔ Small # of Masters/Slaves per layer**
- **Various dynamic memory type support**
  - **Flexible memory I/F**
- **Secure region support**
  - **For data security**

Design strategy

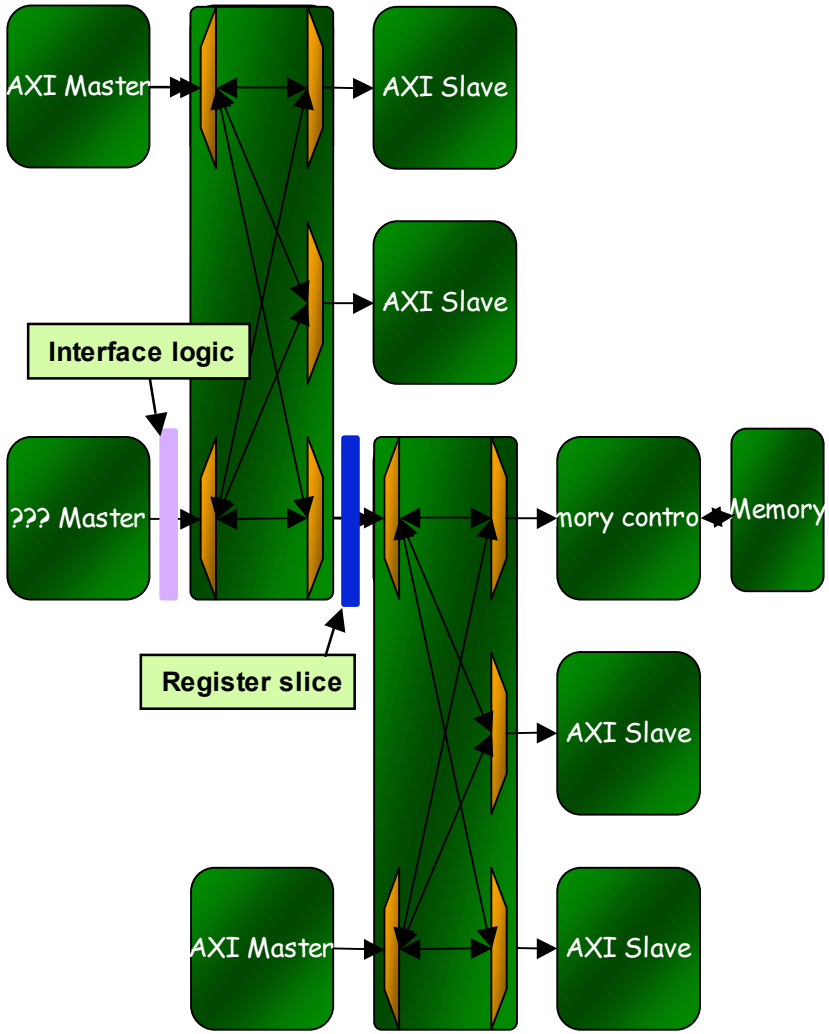
- CPU
  - Most recent ARM processor: ARM1176
- Bus
  - Hybrid style
    - Subsystem: AHB
    - Main backbone: PL300 based on AXI (AMBA3.0)
- Memory controller
  - AXI compliant memory controller
- Design methodology: System2RTL design tool
  - Architecture exploration
  - RTL verification
  - RTL integration

ARM1176 overview

- 4-ported AXI interface
  - Data, instruction, DMA, peripheral I/F
- Low power support
  - IEM (Intelligent Energy Manager): DVS support
  - Power down mode for static leakage reduction
- TrustZone support
  - Trusted computing for critical system function and/or copy right protection
- SIMD media extension
- Separated load-store and arithmetic pipelines
- Pipeline depth: 8 stages
- Branch prediction
- Optional TCM, VFP, …

AXI overview (I)

- **Consistent point-to-point I/F**
  - Systems can be constructed hierarchically – PL300
- **Arbitrary MxN full crossbar**
  - Extended to support partial crossbar switch
  - Round-robin / Fixed-priority
- **Register slicing**
  - Critical paths in the design can be addressed using the register Slice component to pipeline channels
- **Backward compatibility for AHB/APB**
  - Appropriate wrappers are supported



AXI overview (II)

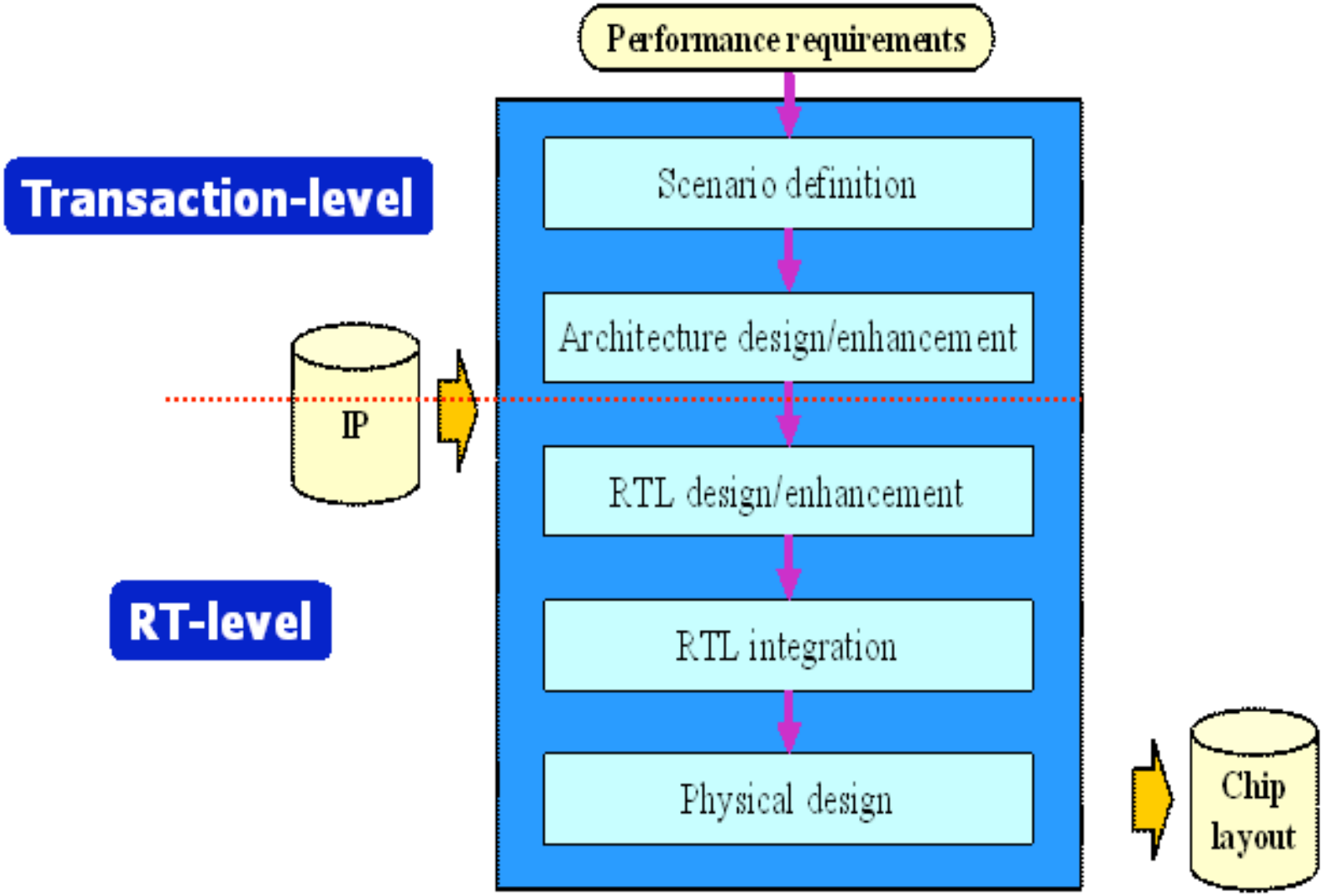
- Only start address issued for bursts
- Separate address, read and write data channels
  - 5 separate channels (RA, WA, RD, WD, WR)
  - Wire counts: 184 ~ 204 when 32-bit address and data
- Multiple outstanding address issue & out-of-order completion
  - In-order completion of transactions with same ID
- Data interleaving
  - For transactions with different IDs in write mode (AWID)
- Optional extensions to cover signaling for low-power operation
- Atomic access
  - Exclusive access, Locked access
- Memory controller
  - Transaction scheduling
  - QoS

AXI overview (III)

Components		Descriptions
Protocol conversion	AxiToApb	Covert from AXI to APB (TZ-aware converter)
	AxiToAhb	Convert from AXI to AHB
	AhbToAxi	Convert from AHB to AXI
Data width conversion	ExpanderAxi	Narrower master to wider bus
	FunnelAxi	Wider master to narrower bus
	DownsizerAxi	Wider bus to narrower bus
	UpsizerAxi	Narrower bus to wider bus
Clock domain crossing	RegSliceAxi	Break long timing paths (can be bypassed)
	SyncUpAxi	Connect a primary AXI to a higher clock frequency (1:n)
	SyncDnAxi	Connect a primary AXI to a lower clock frequency (n:1)
	AsyncAxi	asynchronous bridge (n:m)
Trust zone	TPprotCtrl	TrustZone protection controller
	TAMemAdapAxi	Blocks non-secure access to secure region
	TZIC	TrustZone interrupt controller

Design methodology

❑ Overall design flow



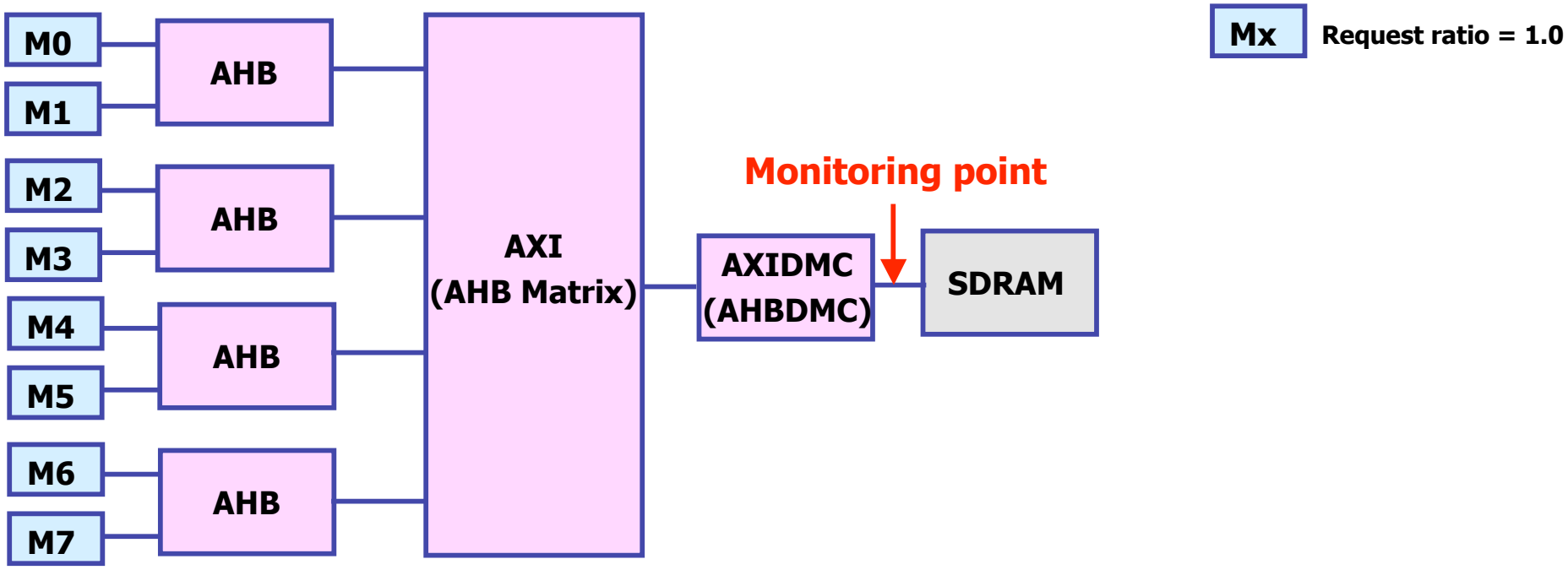
Design methodology (II)

- **Clock speed consideration in system-level**
  - **Legacy IPs**
    - **Pre-characterized by IP designer**
  - **New IPs**
    - **1st stage: target frequency**
    - **2nd stage: estimation from behavioral synthesis tool**
    - **3rd stage: Same as legacy IPs**
  - **Flexible IPs like bus**
    - **Multi-dimensional characterization**
      - **# of masters**
      - **# of slaves**
      - **WriteIssuingCapability**
      - **....**
    - **Essential to consider register slicing**
    - **Other metrics (e.g. power, gate count) can be considered in a similar fashion**

AHB vs. AXI

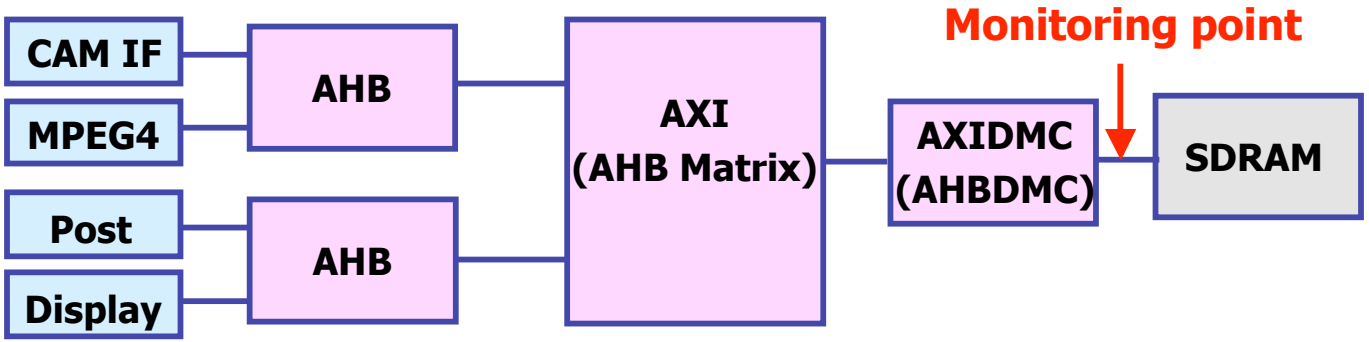
- To justify the use of AXI
- Metrics
  - Memory bandwidth utilization
    - Data transfer / cycle
    - The higher, the better
  - Clock speed
    - Good enough if it is higher than the target clock speed
- Comparison method
  - Synthetic workload
    - To see the performance when the workload is maximum
    - AXI: AHB master + AHB2AXI + PL300 + PL340
    - AHB: AHB master + Bus matrix + DMC
  - Baseline architecture
    - To see the performance in a real situation

AHB vs. AXI – Synthetic case



- **Read operation**
  - AXI outperforms AHB upto 80%
- **Write operation**
  - AXI slightly outperforms AHB
    - Write bank interleaving is not supported

AHB vs. AXI – baseline architecture (I)



- Running scenario
  - MPEG4 CIF encoding + VGA display
- Memory map (by controlling data location)
  - Best: Maximize bank interleaving
  - Worst: Minimize bank interleaving
  - Typical: bank selected by addr[12:11] ➔ close to random

AHB vs. AXI – baseline architecture (II)

- Execution cycles normalized to Best AXI

	AXI	AHB
Best	1	1.25
Typical	1.04	1.48
Worst	1.28	1.59

- AXI outperforms AHB
- AXI is less sensitive to memory map
- AXI can be better by enhancing memory controller (adding write bank interleaving)

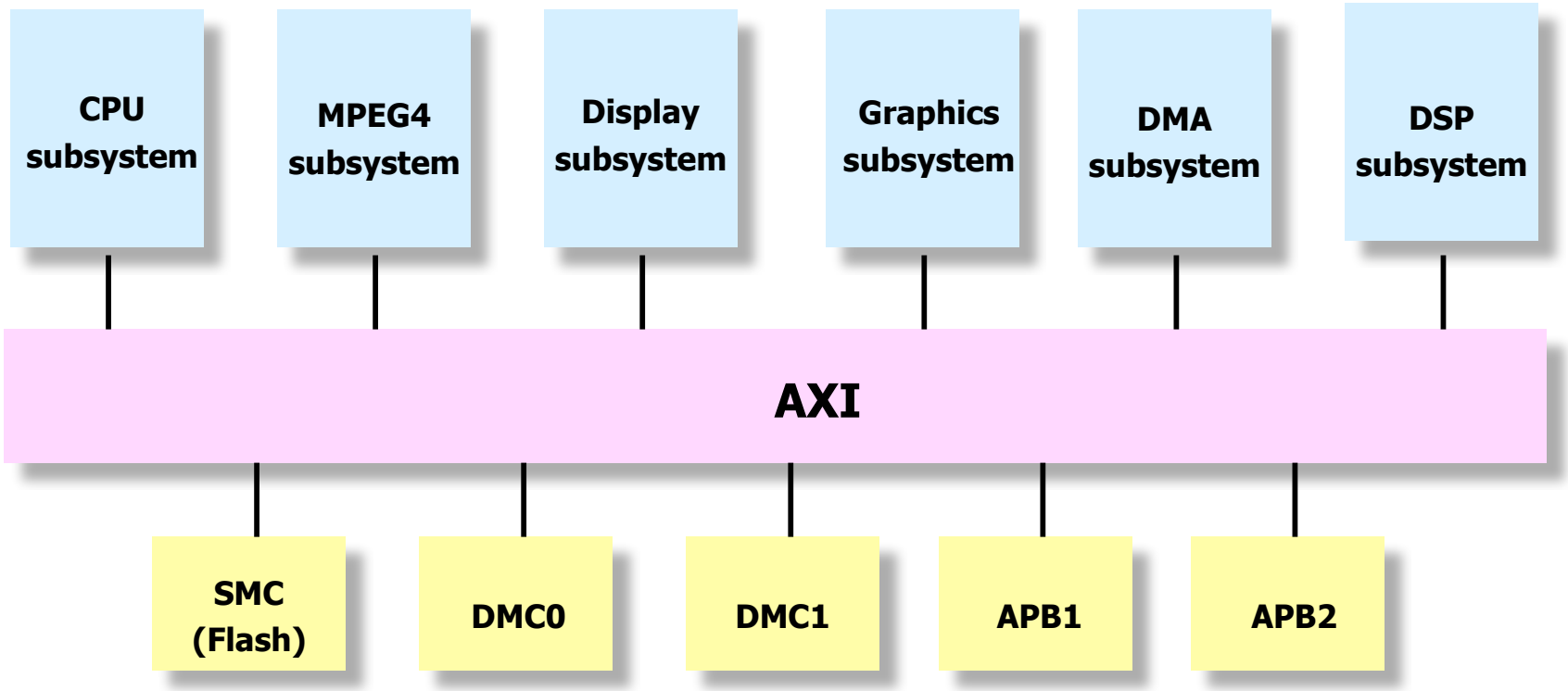
Memory controller enhancement (I)

- TLM environment
  - Performance bottleneck identification
  - Easy to quantify the enhancement ratio
- RTL environment
  - Need to check any side effects after modification
  - Used SpecMan eVC with scoreboarding

Memory controller enhancement (II)

- **Write bank interleaving**
  - Hiding the row activation time when a series of transactions accessing different banks are in a queue
  - 41% performance improvement
- **Write data FIFO merging**
  - Write data FIFO is necessary to handle multiple outstanding transactions
  - A single transaction divided into several transactions when the write data FIFO is full
  - Can be merged when the space becomes available

New architecture of Mobile-AP



Worst case scenario

- Define the worst case scenario
  - The most performance demanding scenarios
  - Each scenario has detailed information on each IP behavior
- Objective
  - Analyze the # of frames at the target clock speed for each scenario

Scenario		Description	
1	MPEG4 decode	VGA, 30fps	deblock, display, post
2	MPEG4 encode	VGA, 30fps	camera, display

Scenario detail

MPEG4 decoding scenario

- IP behavior: Action, parameter, data dependency
- Perf. requirement: throughput, response time

IP	Action	Parameter	Throughput	Resp. time
CAMIF	IDLE			
MPEG4	decode	picture_size = VGA	30 fps	
Deblock	deblock	size = VGA	30 fps	
Rotator	IDLE			
Post	Color space conversion	Input = 4:2:0 YCbCr, VGA Output = 24bpp, VGA	30 fps	
Display	LCD control	size=VGA	60Hz	real-time constraint

Architecture exploration

- **Layer optimization**
  - **Original**
  - **Alternative 1**
    - **Deblock is moved from display layer to MPEG4 layer**
  - **Alternative 2**
    - **Display is assigned to a separate layer**
- **Burst length optimization of display module**
  - **Default setting burst 4 (in previous design)**
  - **Alternative 1**
    - **Burst16**
    - **Trade-off between throughput and latency**
    - **Need to satisfy the given real-time constraint**

Experimental results

TLM simulation

- Picture and display size: VGA
- Clock speed: 133MHz

Architecture	Architecture description	Display BL=4	Display BL=16
Original	Layer1: CAMIF, MPEG4 Layer2: POST, POST, DISPLAY, DEBLOCK, Rotator	RTC miss	satisfied
Alternative 1	Move DEBLOCK to layer2	RTC miss	satisfied
Alternative 2	Create layer3 only for DISPLAY	Not necessary	Not necessary