1D-19 : Design and Implementation of a Duplex AMBA-TMS Transducer
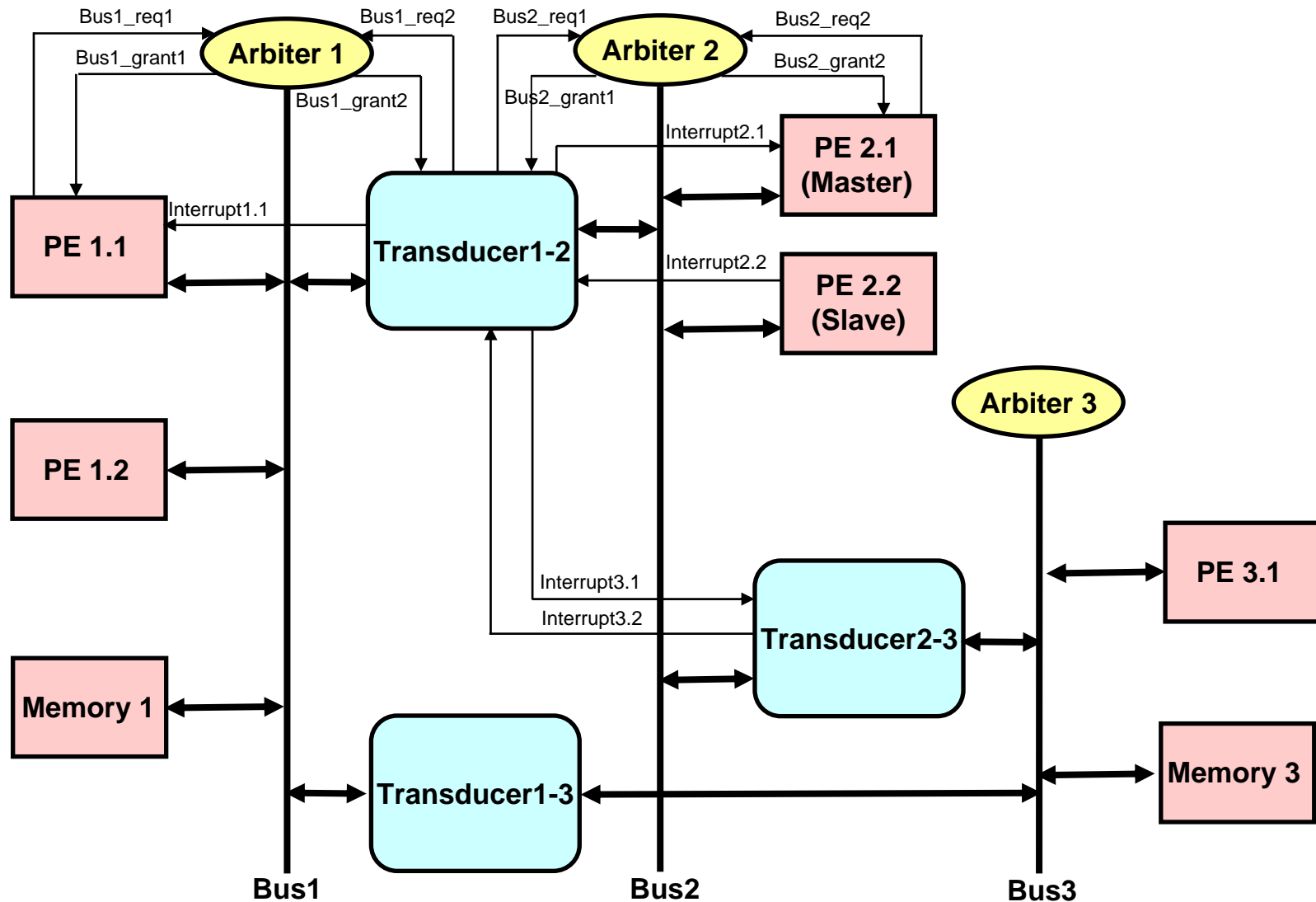
Hansu Cho

Samar Abdi

Daniel Gajski

January, 25 2006

Center for Embedded Computer Systems
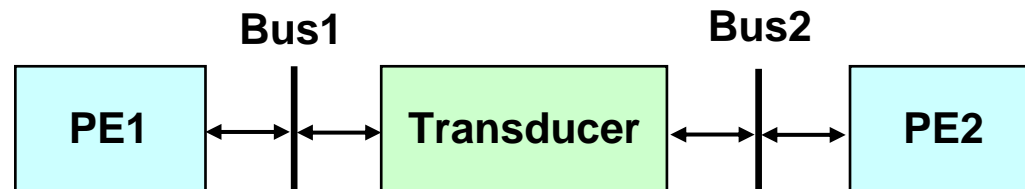
University of California, Irvine

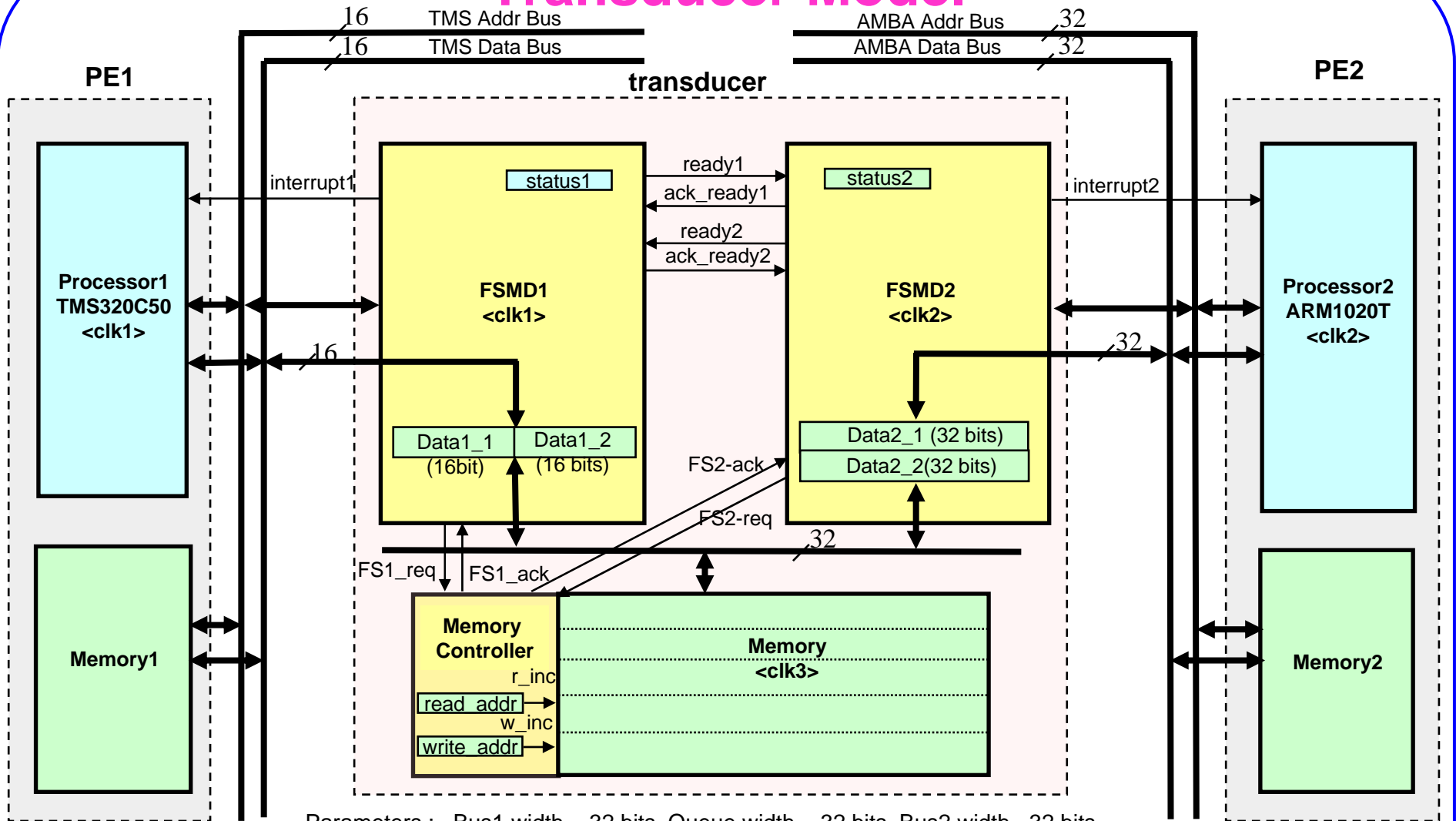# General System Model

Copyright ©2006 Hansu Cho

Problems

- **Different cores in system may have incompatible interface**
  - Different clock speed
  - Different control signals
  - Incompatible ports

- **Need interface logic (transducer)**
  - Translates between protocols of Bus1 and Bus2

```
            Bus1                      Bus2
 ┌──────┐    |    ┌────────────┐       |    ┌──────┐
 │ PE1  │◄──►│◄──►│ Transducer │◄──►   │◄──►│ PE2  │
 └──────┘    |    └────────────┘       |    └──────┘
```

- **Need synthesis algorithm for transducer generation**

Transducer Model

PE1

Processor1
TMS320C50
<clk1>

Memory1

transducer

16 — TMS Addr Bus
16 — TMS Data Bus

interrupt1

status1

FSMD1
<clk1>

16

Data1_1
(16bit)

Data1_2
(16 bits)

FS1_req   FS1_ack

**Memory
Controller**

r_inc

read_addr

w_inc

write_addr

ready1
ack_ready1

ready2
ack_ready2

status2

FSMD2
<clk2>

Data2_1 (32 bits)
Data2_2(32 bits)

FS2-ack

FS2-req

32

**Memory
<clk3>**

AMBA Addr Bus — 32
AMBA Data Bus — 32

interrupt2

32

PE2

Processor2
ARM1020T
<clk2>

Memory2

Parameters :   Bus1 width = 32 bits, Queue width = 32 bits, Bus2 width =32 bits
sBus1 = TMS320C5X
Bus2 = AMBA
PE1 = Dummy custom processor (even parity checker)
PE2 = Dummy ARM1020E (one's counter)
clk(clk1 = 50 Mhz, clk2 = 70 Mhz, clk3 = 30 Mhz)
FIFO Memory = Synchronous SRAM
Status bits : FSMD1_ Ready, Data2_Ready,Q_Full

CECS

Conclusions

- **Results**
  - Implemented a transducer for TMS and AMBA protocols
  - Demonstrated a working transducer on FPGA

- **Benefits**
  - Bus protocol to bus protocol translation
  - IP protocol to bus protocol translation

- **Contributions**
  - Provided general communication interface for multi-core systems
  - One transducer for all
  - Transducer generation algorithm