#### **Configurable SID-based Multi-core Simulators for Embedded System Education**

Chung-Wen Huang,



Wei-Kuan Shih,

Yarsun Hsu,

Jenq Kuen Lee (jklee@cs.nthu.edu.tw) National Tsing-Hua University, Taiwan. Embedded Software Consortium, Taiwan.





## Outline

- Background
- SID Simulation Framework
- Heterogeneous Multi-core Simulation Platform
- Embedded System Courses/Projects based on SID Framework
- Conclusion



## Background of Taiwan Embedded Software Consortium

- The Embedded Software Consortium, established in February 2004 in Taiwan, is the consortium funded by the Ministry of Education (Other consortiums including EDA, heterogeneous integrations, etc).
  - The ESW Consortium focuses the development of embedded software curriculum.
  - We hope to provide a reference curriculum for universities in Taiwan to develop their embedded program.
  - Developing embedded multi-core curriculum is becoming a focus for this program.



# CourseWare: Multi-core Simulation Tools

- With the emerging of multi-core designs for embedded systems, there is a need of multi-core simulation tools for courseware and class experiments.
- Several issues are considered.
  - 1. To have a sequence of course work based on similar toolkits
  - 2. To be able to configure the architectures with a variety of processor IPs (including MPU and DSP processors).
  - 3. To accommodate ingenious local IPs for experiments.
  - 4. Experiment with multi-core programming models.
  - 5. Experiment with interconnection networks.
  - 6. Experiment with embedded multi-core applications.
  - 7. Learn how to track and debug multi-core programs and performance tuning.
  - 8. Experiment with middleware experiments.



#### **StarIP Programs**

Key Techniques in Chip Systems

Messenger – Distributed Radio Transmitter System



#### **Experiments with Various Parallel Programming Models**

#### • MCAPI

- Released the first spec at 2008 by the Multicore Association
- A standardized API for communication and synchronization
- Focus on high performance and low memory footprint
- Defines three communications type: Messages, Packet channels, and Scalar channels
- CUDA
  - Introduced by NVIDIA at 2006 for GPU architecture
  - C extensions and need compiler support
  - SPMD-like programming model
  - Fine-grained data parallelism and thread parallelism

Application	Application	Application
МСАРІ	мсарі	мсарі
Further abstraction (OS/middl	eware)	
	Interconnect topology	
CPU/DSP Accelerators	CPU/DSP Accelerators	CPU/DSP Accelerators
Source: http://ww	Software ww.multicore-associat	tion.org/



Source: NVIDIA Corp., "NVIDIA CUDA Compute Unified Device Architecture Programming Guide V2.0", 2008

OpenCL

- Released the first spec at 2008 by Khronos Groups with many industry-leading companies and institutions
- Target for heterogeneous multicore environment
- For general purpose applications
- OpenMAX
  - Released the first spec at 2007 by Khronos Groups
  - Defined three layers of API interface to abstract software and hardware difference for portability
  - For multi-media applications
  - Lacked of multicore support
  - Streaming RPC
    - ICPP 2008
    - Integration of Remoting with streaming mechainisms





Source: Giulio Urlini, "The OpenMAX Integration Layer d Standard," CELF Embedded Linux Conference, 2007

Source: Neil Trevett, "OpenCL the Open Standard for Heterogeneous Parallel Programming," SIGGRAPH Asia, 2008

## **SID Multicore Simulation Milestones**

#### • SID Origins (<2001)

Embedded

Consortium

- SID is an open source framework for building computer system simulations.
- The simulation framework is comprised of a set of loosely coupled components ( in C, C++ or TCL).
- SID was first released at 2001 Embedded System Conference.
- SID in the Public Domain (<u>http://sourceware.org/sid/</u>)
  - 100+ components are source code available in public domains.
    - such as processors, memory, DMAs, LCDs, and peripherals.
  - 10+ architecture description files available for references.

#### SID Components of Taiwan Dual-core Platform (2007-2008)

- 5-way issues VLIW DSP component (NTHU, ITRI STC)
- Virtual I/Os of Video Camera and Network Interface. (NTHU)
- SID Multicore Simulation (2008~2009) (NTHU)
  - Configurable multi-core simulation with one MPU and four DSPs.
  - Multi-core GUI development environment.
  - OCP compliant interconnect implementation.
  - Multi-core data profiling and tracing.



# **Comparison of ESL Simulation**

Items	<b>CoWare™</b> Architecture Designer	<b>SID</b> Simulation Framework	<b>QEMU</b> open source processor emulator
Language	C++/SystemC	C++/C/TcI	С
IP-based (Configurabl e)	Yes	Yes	No
Free	No	Yes	Yes
Source code	No	Yes	Yes
Scheduler/ Clock Gen	Central clock source	Individual time quantum	none
Performance Accuracy of Simulation	High (Most IPs are cycle-accurate.)	Middle (Processor IPs are cycle-accurate.)	Less (Most IPs are functional-accurate.)

Embedded

Consortium



# **SID Simulation Framework**



- According to the types of modeling targets, SID components can be classified into hardware, software, and system components.
- The most important system components are the main controller (*cfgroot*) and the *scheduler* component.
  - The *cfgroot* controls the compositions of components and the main execution of simulation.
  - The scheduler then arranges the execution sequences of components.
    Workshop on Embedded Systems Education, 2009

9

# **Layers of Simulation Models**





## **Configurable Interconnection Model**

 For the original SID framework, the bus mechanism is emulated by look-up tables and the memory addresses are mapped to the read/write functions of components.

Consortium



- We add new data communication flows of the interconnection networks.
  - An new interconnection adaptor (IA) accepts the bus control and transforms data packages.
  - The new interconnection component is a package of mathematical models with the statistics of package transmission history and a probabilistic analysis of communication predictions.
  - The interconnection model now can be configured as bus, crossbar, and on-chip network.

### **Heterogeneous Multicore Simulation Platform**



- Based on Open64 compiler
- VLIW DSP compilers for distributed register files
- PALF scheduling policies for ILP (CPC 2006)
- GRA scheme for distributed register files (CPC 2007)
- SIMD compiler optimizations + intrinsics/extrinsics
- Copy propagations for distributed register architectures (LCPC 2006)
- Register spills among distributed register banks (CPC 2009)



#### **PACDSP Compiler**



#### **SID Framework for Embedded System Course**

- The tool is developed to hope to be able to provide labs for several graduate embedded courses.
  - 1. Toolchain for Embedded Software (ES-Y04-2)
  - 2. Embedded Compiler Design (ES-Y05-1)
  - 3. Embedded Multimedia Design (ES-Y08-1)
  - 4. Embedded Hw/Sw Co-design and Analysis (ES-Y08-3)
  - 5. Heterogeneous Multi-cores Course (ES-Y09-1)

Fie Edit Navigate Search Bun Pro	ect Target Window Help										
6			E Coder »								
1 1 · 2 · 2 · 3 · 2 · 3 ·	📾 🕸 🖓 🎖 🖓 🖓 🖉	in - je v 😳 🗇 🖉 🖉 🖾 🔛 🔛 🔛 🔛									
🚆 Proje 22 Class Navig 📟 🗖	dema2_in.conf		= 0 🗄 Outine 31 - 0								
0000000	new hw-mailbox pac04_mailbox		An outline is not available.								
▼ 🍃 demo2	new hw-cpu-rds32hf cpu				$\sim$			c			
D O Binaries	# bus section				Sr	nang	chote	s ot a	VOCUTIO	n	
Debug	Finew hw-akbc-atc020 AHB new hw-IN-basic AHB				J	iaps	ποι	5010	ACCUIR		
V 🌝 JPEG_DualCore_240_320	# some default attribute					•					
D 👝 (PEG	set cpu step-insn-count 1 #10000	and the second	_								
PEG_DualCore.com	set pac01 step-insn-count 1		me	e		Calls Self InsC	Self CycC Se	If InsC/Call Self CycC/Call	Total Call InsC Total Call CycC	CPI	Time Percentage
😁 Protie	set pacH2 step-insn-count 1		con	mpress onepass		20 11,365,540	8 943,477,919 !	568,277.40 47,173,895.95	795,781.80 66,067,947.90	83.01 40 22%	
🗢 🎉 maibox	set paci3 step-insn-count 1		0	fanzy unsample		320 4 550 400	377 695 015	14 220 00 1 180 296 92	14 220 00 1 180 295 92	83.00 16.10%	
Debug	set pac94 step-insn-count 1 #1000			rah convert		320 3.942.09/	127 196 045	12 319 00 1 022 497 64	12 119 00 1 022 497 64	83.00 13.00	
V Chinux_Demol_Andes	art cou engine type pbb			rigo_converc		320 3,342,000	100 500 445	2022,407.04	7,333,03	03.00 03.00	
D 🔝 Andes, prog. c	# set cpu endiam big		CP	pixel_rows		320 2.320.320	0 192,589,445	7.251.00 601.842.02	7,332.93 608,651.30	83.00 8.21%	
pac_linux.conf	# attribute section set Memory size 0x02000000	(240,320) : 24bpp : 8GR	9_	_idct_islow	2		Pro	file		X 01 6.37%	
þ 🇀 Profile	set Memory2 size 0x04000000	ظ <sup>ار</sup> pickki/20losilhort – [106s23]		ode_mcu	PAC Profile					09 0,03%	
	set LCDC bus-endian 2	地球口 编辑目 枪成了 成面面 建藻目 脱消日	the	ne_clean_d	Eunctions	Ovclos	Cyclos(%)	Called	(96)	00 3.52%	
D O Binaries	and the character can it way a can it is	[IN.D] Step:49329319, Burst mode, normal read. [IN.b] Step:49329319 Master:[ICD7] Slave:[memory] Burst read Read 57600.	data each bas Abytes	fill_bit_buffer	crt1 start	20	0.0019579623907	02 1	0%	05 1.74%	
D Go 5_core	# pin connections	[IN.h] Step:49359524. Burst mode, normal read.	data, cacil ilas voytes.	iset		20	0.0010370023007	721	078	01 1.26%	
Debug	contect-pin Timer tw2 intr -> INTC interrupt-son contect.pin Timer tw2 intr -> INTC interrupt.son	[IN.h] Step:49359524 Master:[LCDC], Slave:[memory]. Burst read. Read 57600	data, each has 4bytes. mo	сру	main	22	0.0020436486187	12 2	0%	0.82%	
😓 Profile 🗸 🗸	contect-pin INTC fast-interrupt -> cpu huß contect-pin INTC interrupt -> cpu hul	[IN.h] Step:49387559. Burst mode, normal read. [IN.h] Step:49387559 Master:[ICDC] Slave:[memory] Burst read Read 57600.	data each has thetes	fread	.BB2_main	15	0.0013933967855	26 2	0%	0.74%	
	<pre>contect-pin INTC_pac01 fast-interrupt -&gt; pac01 f</pre>	[IN.h] Step:49307559 master. [Debe], Stave. [memory]. Burst read. Read 57000	data, cacil has objects.	make d derived	.BB3 main	9	0.0008360380713	15	0%	03 0.19%	
Kill Target Manager 11	contect-pin INTC_pac02 fast-interrupt -> pac02 f contect-pin INTC_pac03 fast-interrupt -> pac03 f	[IN.h] Step:49417778 Master:[LCDC], Slave:[memory]. Burst read. Read 57600	data, each has 4bytes.	unsample	BB5 main	15	0.0013933967855	26 2	0%	13 0 18%	
\$	connect-pin INTC_pac04 fast-interrupt -> pac04 f	[IN.h] Step:49446417. Burst mode, normal read. [IN.h] Step:40446417 Montary [LCDC] Slave:[memory] Burst read Read 57600.	data anch has thetas	es virt sarrav	DDE main	20	0.0026020004520	17	0%	11 0.09%	
0	contect-pin pac01 mailbox interrupt-to-pac -> IN contect-pin pac02 mailbox interrupt-to-pac -> IN	[IN.h] Step:49470690. Burst mode, normal read,	uata, cacil llas abytes.	ass_vic_surray	.DD0_IIIdili	29	0.0020939004320	11	076	0.07%	
VEP [192.168.0.99][5_care.conf_IN.cor	contect-pin packs_mailoux interrupt-to-pac -> in contect-pin packs_mailoux interrupt-to-pac -> IN	[IN.h] Step:49470690 Master:[LCDC], Slave:[memory], Burst read, Read 57600	data, each has 4bytes.	ess_data_contex	Ackermann	321270	29.843772352406	8% 21418	30%	21 0.07%	
	ronart nin nar81 million internet to oir . 18	[IN.h] Step:49498249. Burst mode, normal read.	data anch has divites	hum_decode	.BB2_Ackermann	128508	11.937508940962	7% 5295	12%	17 0.06%	
	Ponne St. Broblems Bronastias Datus	[IN.h] Step:49496249 master.[DDDC], Slave.[nemoly]. Burst read. Read 57000	uata, caci nas abytes.	_ycc_rgb_table	.BB4 Ackermann	1452	0.1348808088389	66% 121	0%	0 0.05%	
	pac Score adx (Cross Platform Application) Milinferior/hs	[IN.h] Step:49528501 Master:[LCDC], Slave:[memory]. Burst read. Read 57600	data, each has 4bytes. 9-	_read_scanlines	RB5 Ackermann	01101	8 4710164170208	6% 2/12	8%	18 0.05%	
	Start one Line	[IN.h] Step:49558725. Burst mode, normal read.	ind	d_specmb	DOJ_ACKEIIIIdilli	74100	0.4710104179290	10500	370	13 0.03%	
	the results	[IN.h] Step:49536725 Master.[DCDC], Slave:[memory]. Burst read. Read 57600	uata, each has 4bytes.	_sarray	.BB7_Ackermann	74102	0.8835059067390	2% 10580	1%	03 0.03%	
		[IN.h] Step:49588158 Master:[LCDC], Slave:[memory]. Burst read. Read 57600	data, each has 4bytes. eg	g_main	.BB8_Ackermann	52930	4.9168327905278	7% 10586	5%	13 0.03%	
		[IN.h] Step: 49614501. Burst mode, normal read.	state and have the state	are_range_limit_t	.BB10_Ackerman	n 406942	37.8021116463819	9%	38%	0.03%	
		invinj Step:49014501 Master:[LCLC], Stave:[memory], Burst read. Read 5/600	uata, each nas 4bytes.	new file xsputn	-	46: 6,05	8: 504,784	131.70 10,973.57	264.04 22.049.87	83 33 0.02%	
Andes Status			N V	View Call View Tir	meline View Chart	View					

## **SID Framework for Embedded Courses**

- The on-going courses and lab modules on the SID simulation framework at NTHU EE/CS.
  - Category of System Tool: (Prof. Jenq Kuen Lee)
    Embedded Compiler Design (ES-Y05-1), Special Topics on Advanced Programming Languages
    - Streaming RPC on Multi-core Platform
    - Enable the Software Cache on DSP Processors
  - 2. Category of Architecture: (Prof. Yarsun Hsu)

Advanced Computer Architecture, Embedded Hw/Sw Co-design and Analysis (ES-Y08-3)

- Comparisons of SID Interconnection to SystemC TLM IP
- 3. Category of Applications: (Prof. Wei-Kuan Shih and Prof. S. H. Lai) Embedded Multimedia Design (ES-Y08-1), Toolchain for Embedded Software (ES-Y04-2), Special Topics on Embedded System Designs
  - Face Recognition Applications
  - Belief Propagation Method for Stereo Vision

## **Streaming RPC on Multi-core Platform**

 – Key environment setting: MPU, DSP, Interconnection, Interrupts, and OS

### – Project goal:

Consortium

Experiment with DMA effects with streaming remoting programming model on SID multi-core platforms



\*\* Enabling Streaming Remoting on Embedded Dual-core Processors, Kun-Yuan Hsieh, Yen-Chih Liu, Po-Wen Wu, Shou-Wei Chang, Jenq Kuen Lee, ICPP 2008, Portand, Oregon, Sep. 8-12, 2008.

#### **Experiments with Software Cache on DSP Processor**

- Key environment setting: MPU, DSP, Interconnection, DMA, and Local/External Memory
- Project goal:
  - 1. Implement Software Cache on the non-data-cached PAC DSP.
  - 2. Experiment with DMA to accelerate the software cache.

Name	Description
pCore_sw_cache_read()	Return the data value from the cache.
pCore_sw_cache_write()	Write the value into the cache.
pCore_sw_block_read()	Fast data block read function reducing
	address lookup overhead.
pCore_sw_block_write()	Fast data block write function to
	the external memory.
pCore_sw_block_trans_done()	Indicate that the block transaction
-	is done.
pCore_sw_cache_lock()	Function that locks the specified
-	cache line.
pCore_sw_cache_unlock()	Function that unlocks the specified
• 0	cache line.





#### **Belief Propagation Method on Embedded Multi-core Processors for Stereo Vision**

- Key environment setting: MPU, DSP, DMA, and Local/External Memory
- Project goal:

Parallelize the belief propagation method on multi-core PAC DSP environment

• Strategies:

Consortium



Building the data pyramid from lowest level data layer



(a) Original image pair.



(b) Ground-truth.



(b) Result of parallelized BP.

#### Experiments with SID Interconnection and SystemC TLM IP

- Key environment setting: Processor, Interconnection, Memory, and CoWare<sup>™</sup> tools
- Project goal:
  - Evaluate performance variations of benchmarks in SID and CoWare<sup>™</sup> SystemC IPs.
  - Evaluate accuracy variations of benchmarks in SID and CoWare<sup>™</sup> SystemC IPs.

#### • Strategies:

• Trace the accessing patterns of SID interconnections and compare to the SystemC TLM model.

#### **Feedbacks and Experiences**

- Virtual I/O components provide the graphical experience to attract students' interests on the multimedia projects.
- Embedded multi-core platforms can be configured for embedded multi-core compiler testbed.
- The interconnection of the multi-core platform in our testbed can reflect different topology and communication types.
- For debugging multi-core applications, the multi-core IDE can provide a source-code-level debugging.
- For a difficult bug or a hardware bug, students can use *trace unit component* to record the detailed activities of the whole system.
- Bugs and version fixes slow down student project progresses.
- TAs efforts are needed to help move lab. forwards.

## Conclusion

- We developed an embedded multi-core platform based on SID framework.
- It included MPU, DSP, interconnection, and trace unit. .
- It accommodated ingenious local IPs such as MPU and DSP.
- It's useful for a variety of experiments and lab modules with embedded multi-core topics.
- Several courses are designed based on this tool for lab experiments.

