Heterogeneous Architecture with Configurable Coprocessing Datapaths

Chein-Wei Jen

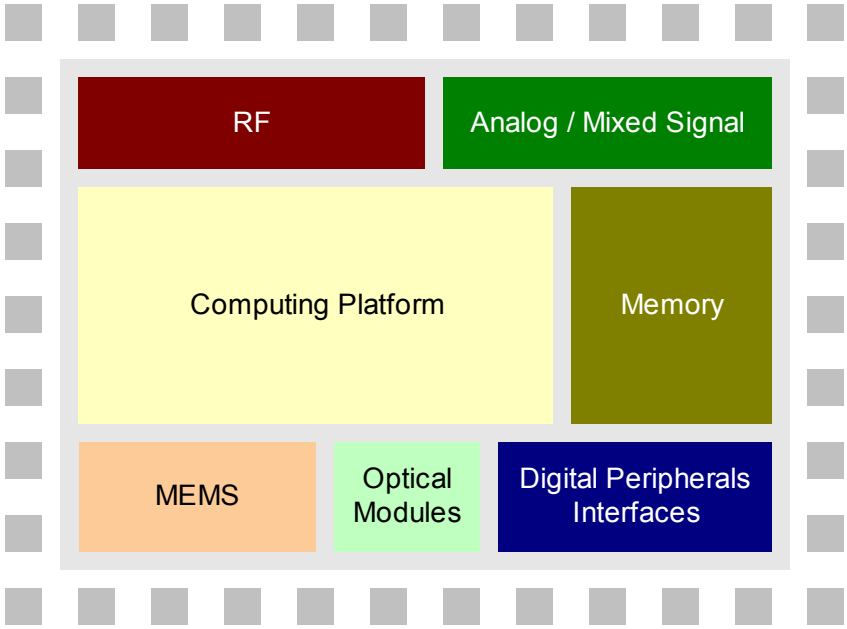
VLSI Signal Processing Group
Department of Electronics Engineering
National Chiao Tung University, Taiwan

VLSI Signal Processing Group
Chein-Wei Jen cwjen@ee.nctu.edu.tw
Department of Electronics Engineering, National Chiao Tung University

Outline

- ***Introduction***
- Coprocessing Datapath
- CASCADE Design Environment
- Design Examples
- Conclusion

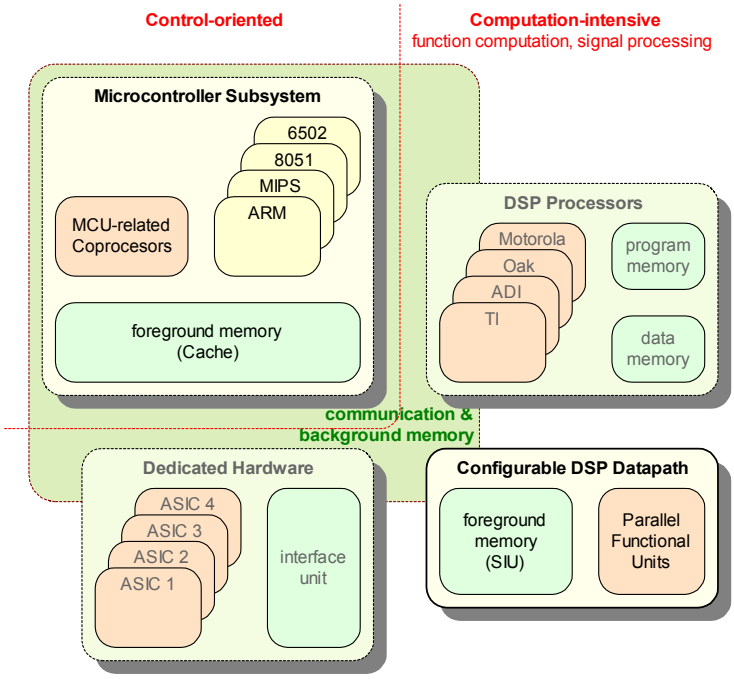
System on Chip



- Huge integration
- Heterogeneous

VLSI Signal Processing Group
Chein-Wei Jen cwjen@ee.nctu.edu.tw
Department of Electronics Engineering, National Chiao Tung University

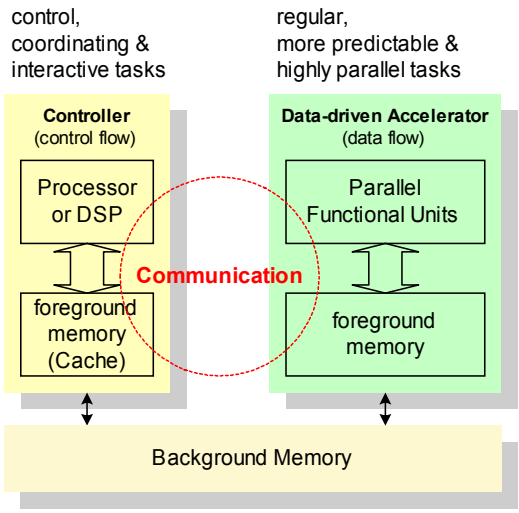
Computing Platform



- Heterogeneous architecture
- Different computing models
- Performance, flexibility,design availability and power consumption

Master-slave Coprocessing

- Coprocessing with two different computing models
- Task partitioning
  - granularity
  - SW/HW
- Interfacing
  - data communication
  - synchronization
    interrupt, semaphore

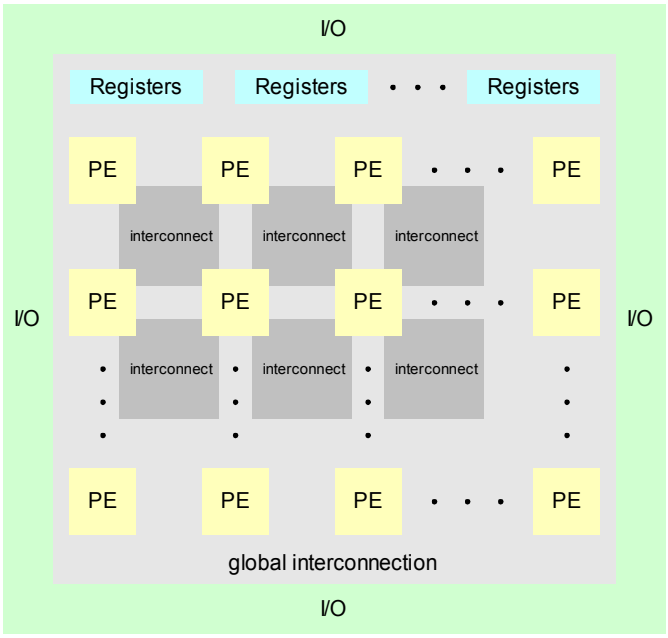


Department of Electronics Engineering, National Chiao Tung University
VLSI Signal Processing Group
Chein-Wei Jen cwjen@ee.nctu.edu.tw

What is 'Configurable'?

- Simple context applying to a basic architecture to change the functions and/or interconnections
- ***Programmable*** : by software, instruction
  Configurable : by register data, memory context
- Change at design time - ***configurable***
  Change after shipment - ***reconfigurable***
  Change at run time - ***dynamically reconfigurable***

Configurable Logic



- PE granularity
  - # of functions
- Interconnection
  - neighbor/mesh
  - crossbar
  - bus
- Context setting
  - download externally
  - in-situ

VLSI Signal Processing Group
Department of Electronics Engineering, National Chiao Tung University
Chein-Wei Jen cwjen@ee.nctu.edu.tw

Outline

- Introduction
- *Coprocessing Datapath*
- CASCADE Design Environment
- Design Examples
- Conclusion

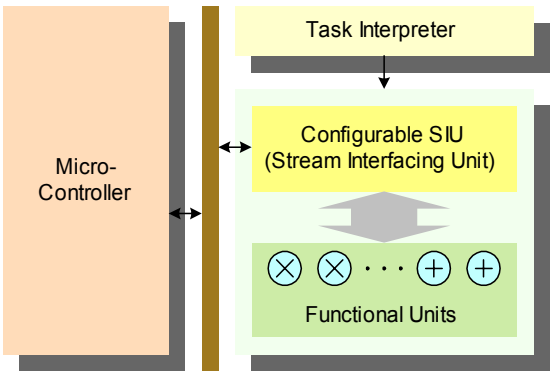
Coprocessing Datapath

- As the complexity of embedded computing grows rapidly, it is common to accelerate critical tasks with hardware and operate under the software control.
- Designers usually use off-the-shelf components or licensed IP cores to shorten the time to market.

- Problems
  - hardware/software interfacing is tedious, error-prone and usually not portable » ***automatic generation***
  - the existing hardware seldom matches the requirements perfectly » ***performance scalable***

VLSI Signal Processing Group
Chein-Wei Jen cwjen@ee.nctu.edu.tw
Department of Electronics Engineering, National Chiao Tung University

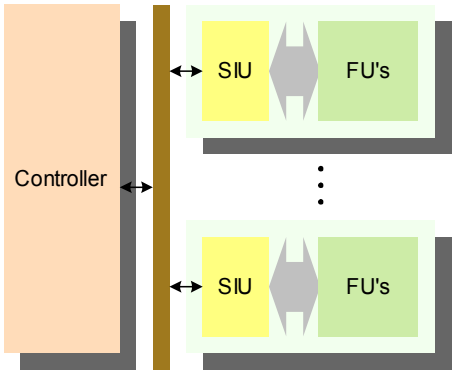
Proposed Computing Model



- The controller coordinates the system tasks and all interfaces.
- Data-driven datapaths function as slave accelerators attached to the controller
- ***Task interpreter*** translates the original control-flow semantics of the dispatched task into dataflow and drives the attached accelerator
- ***Stream interface unit*** (SIU) is an application-specific foreground memory with configurable routing that interacts with the task interpreter
- Data movement is completely under the uC control so ***no explicit data coherence mechanism*** is required
- The computation time of the dispatched task is predictable - easy scheduling for the system

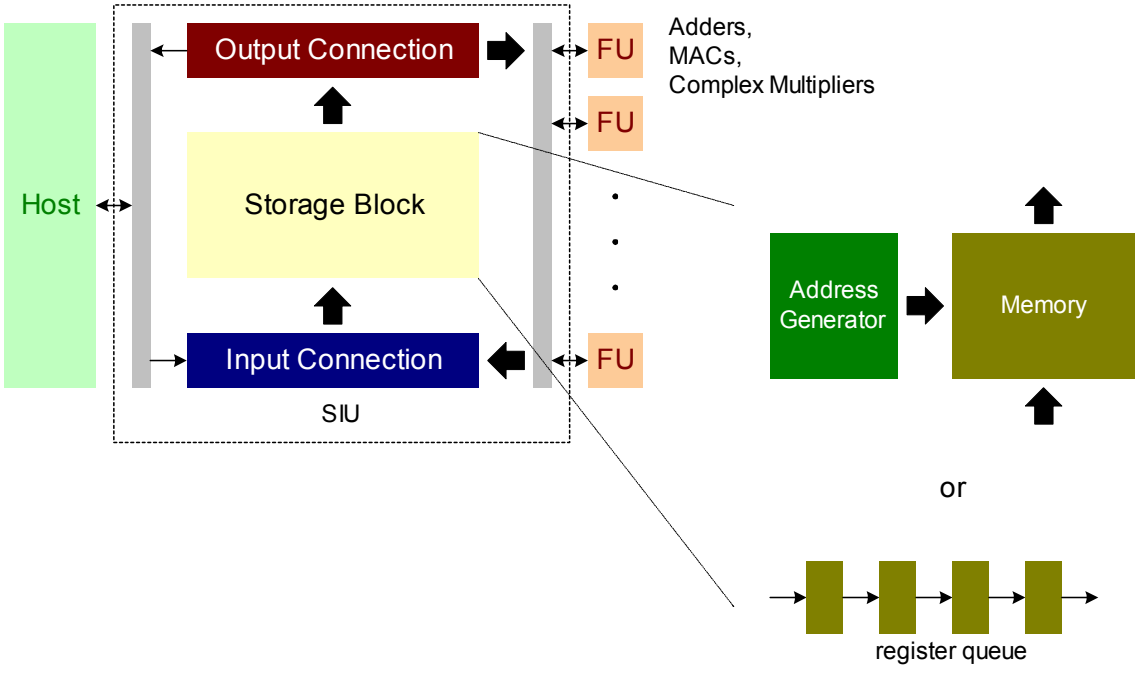
Scalable Performance via Configurable Architectures

- The number of coprocessing datapaths and the parallel functional units inside these datapaths are ***scalable*** to meet the performance requirements.
- The coprocessing datapaths are ***configurable*** for various DSP applications based on the executing algorithms in C/C++ through our generator.
- The performance boost is achieved by
  - control overheads elimination (program flow)
  - reduced load / store operations with specific SIU data generator (data generation)
  - parallel processing via SIMD-like functional units



VLSI Signal Processing Group
Chein-Wei Jen cwjen@ee.nctu.edu.tw
Department of Electronics Engineering, National Chiao Tung University

Stream Interface Unit (SIU)



Outline

- Introduction
- Coprocessing Datapath
- ***CASCADE Design Environment***
- Design Examples
- Conclusion

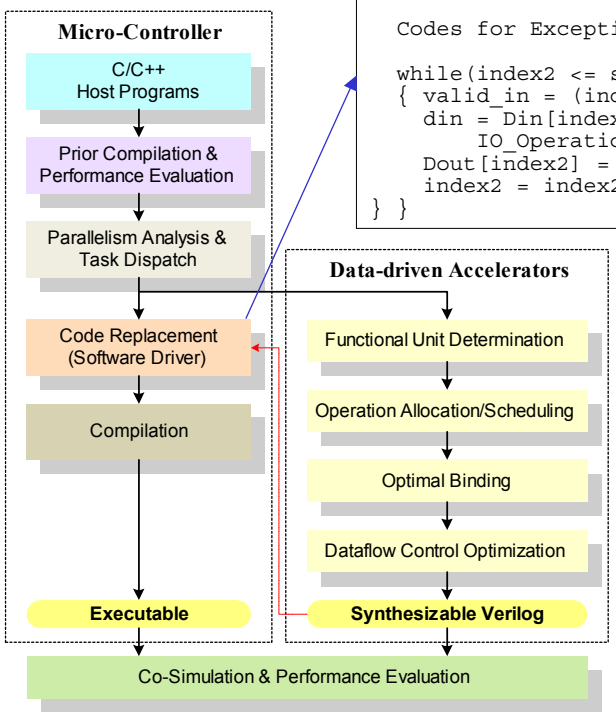
VLSI Signal Processing Group
Chein-Wei Jen cwjen@ee.nctu.edu.tw
Department of Electronics Engineering, National Chiao Tung University

Design Environment – CASCADE

Configurable **a**nd **Sca**lable **D**SP **E**nvironment

- CASCADE ***generates coprocessing datapaths from the executing algorithms*** specified in C/C++ and attaches these datapaths to the embedded processor with the ***auto-generated software driver*** (interface)
- The number of datapaths and their internal parallel functional units are scaled ***to fit the application***
- It seamlessly integrates the design tools of the embedded processor to reduce the re-training and design efforts and ***maintain short development time as pure software approaches***

Design Flow




```
void dispatch_fun(fix *Din, int size, fix *Dout)
{ int   index1=0, index2=0;
  fix   din, dout;
  bool  valid_in, valid_out;

  Codes for Exception Handling

  while(index2 <= size)
  { valid_in = (index1 < size);
    din = Din[index1++];
        IO_Operation (din, valid_in, &dout, &valid_out);
    Dout[index2] = dout;
    index2 = index2 + valid_out;
} }
```

- The data-driven accelerators are synchronized with the host instructions.
- The interfacing software could be scheduled (by compiler, RTOS or manually) easily
- The design is automatic

VLSI Signal Processing Group
Chein-Wei Jen cwjen@ee.nctu.edu.tw
Department of Electronics Engineering, National Chiao Tung University

Front-end Stages

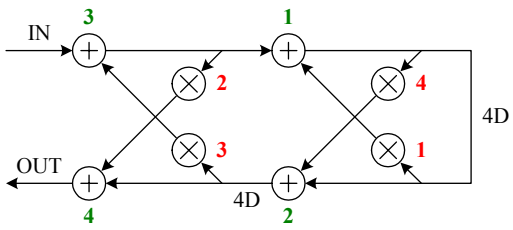
- Functional unit determination
  - MAC
  - (complex) multipliers
  - adders
  - shifters, etc

  For simplicity, the coprocessing datapaths have identical word-length as the host.
- Operation scheduling and allocation
  - estimate the ***maximum allowable cycles*** with an ***acceptable latency*** depending on the ***speedup factor*** by Amdahl's law
  - time-constrained scheduling and allocation (force-directed scheduling)

Optimal Binding

- Calculate the required queuing cycles for each variable

$$D_F\left(U \xrightarrow{e} V\right) = N \cdot w(e) - P_U + v - u$$



- Optimal retiming

Minimize

$$\sum q(U)$$

Subject to

(1) feasibility constraints

$$r(U) - r(V) \le \left\lfloor \frac{D_F\left(U \xrightarrow{e} V\right)}{N} \right\rfloor$$

(2) maximum queue for each variable

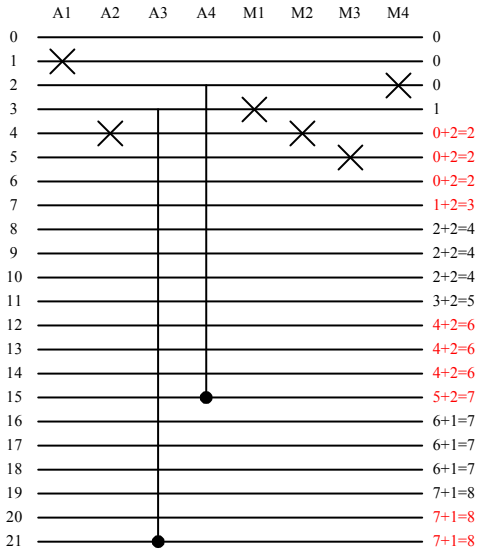
$$D_F\left(U \xrightarrow{e} V\right) \le q(U)$$

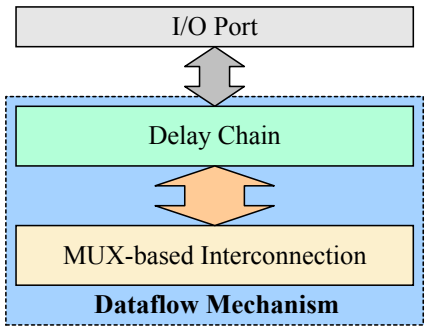
, where

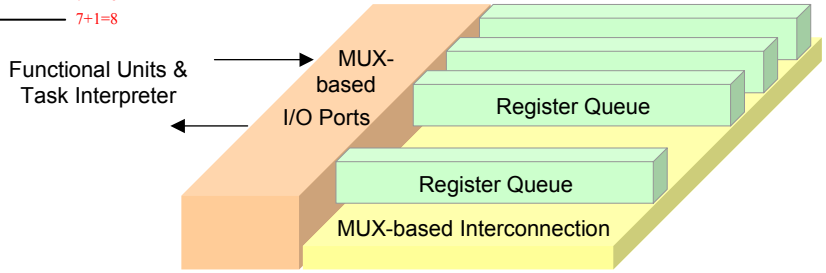
$$D_F\left(U \xrightarrow{e} V\right)$$
$$= N \cdot w_r(e) - P_U + v - u$$
$$= N \cdot [w(e) + r(V) - r(U)] - P_U + v - u.$$

	D_F
A1~A3	0
A1~M1	0
A3~M3	0
A3~M4	13
A3~A4	18
A4~M2	8
A4~A2	13
M1~A2	0
M2~A1	0
M3~A4	0
M4~A3	0

Dataflow Control Optimization



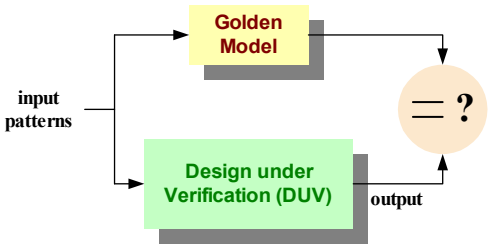




VLSI Signal Processing Group
Chein-Wei Jen cwjen@ee.nctu.edu.tw
Department of Electronics Engineering, National Chiao Tung University

Design Validation

- Design in software
  - algorithm verification & code debugging for a new product
- Verification - equivalence checking (EC)
  - co-simulation
    - equivalence is only guaranteed to some extent that the test suite exercises the design
    - FPGA emulation is required fast simulation



  - formal verification
    - complete EC
    - formal EC between the specification in C/C++ and the auto-generated coprocessing datapath wrapped in the software interface

VLSI Signal Processing Group
Chein-Wei Jen cwjen@ee.nctu.edu.tw
Department of Electronics Engineering, National Chiao Tung University

Formal Equivalence Checker

- Combinational EC algorithms can solve more complex and even unsolved (with distinct I/O sequencing or timing, such as the folded architectures) sequential EC problems by the ***unfolding transform*** with proper ***retiming*** to completely remove the pipeline registers and ***robust register mapping*** for loop synchronization delay elements.
- Our proposed unified algorithm constructs the canonical representation (e.g. ROBDD) direct from the folded architecture with ***implicit*** unfolding and retiming.
- The complex ROBDD construction is only required for the combinational part of the folded architecture, which is much smaller than the flattened unfolded representation. So, the proposed algorithm is ***computation-effective***.
- Identity testing on partial-built ROBDD is possible with our proposed iterative ROBDD construction scheme. Thus, the proposed algorithm is ***memory-effective***.

Comparison - Design Specification

	Modeling	Validation	Domain
Ptolemy (Berkeley)	SDF, DE, etc (heterogeneous)	Simulation	Dataflow
CASTLE (GMD)	C++, VHDL, or Verilog (heterogeneous)	Simulation	
CoWare (IMEC)	C, VHDL, or DFL (heterogeneous)	Simulation	
Vulcan (Stanford)	HardwareC	Simulation	
Cosyma (Braunschweig)	Extended C	Simulation	
Chinook (Washington)	Verilog	Simulation	Control-flow
COSMOS (TIMA)	SDL	Simulation	Communication
POLIS (Berkeley)	Esterel	Formal Verification	
Tosca (Cefriel)	C, VHDL, or Occam	Simulation	Control-flow
SpecSyn (Victoria)	SpecCharts (extended VHDL)	Simulation	
CASCADE	C	Both	Dataflow

VLSI Signal Processing Group
Chein-Wei Jen cwjen@ee.nctu.edu.tw
Department of Electronics Engineering, National Chiao Tung University

Comparison - Implementation

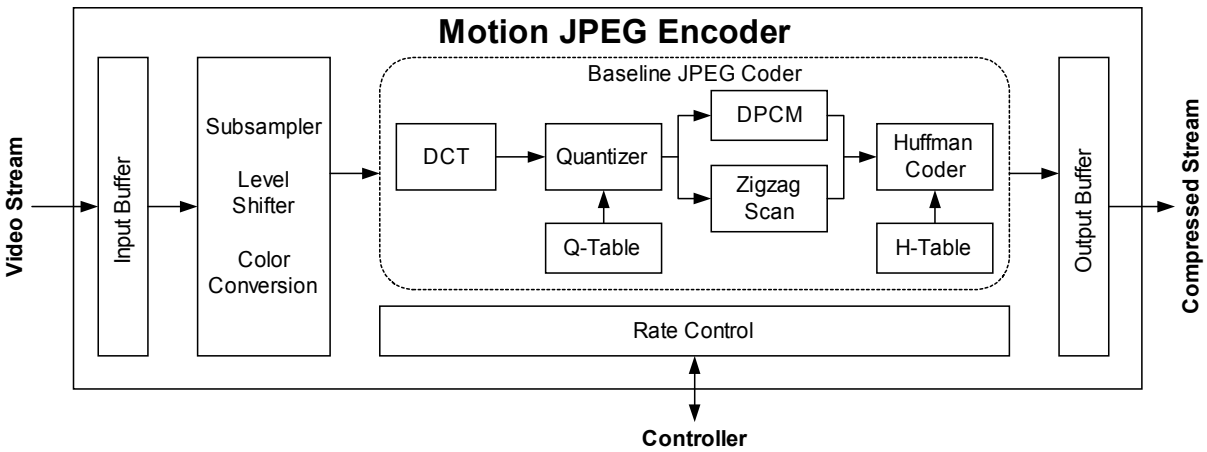
	Target Architecture (# of Processors, # of Coprocessors)	Partitioning	Interface
Ptolemy	1,0		
CASTLE	1,0		
CoWare	n,n	Manual	Automatic
Vulcan	1,n (concurrent)	Automatic	N.A.
Cosyma	1,1(exclusive)	Automatic	None
Chinook	n,n	Manual	Automatic
COSMOS	n,n	Manual	N.A.
POLIS	n,n	Manual	Automatic
Tosca	1,n	Automatic	Automatic
SpecSyn	n,n	Automatic	Automatic
CASCADE	1,n (exclusive)	Manual	Automatic

Outline

- Introduction
- Coprocessing Datapath
- CASCADE Design Environment
- ***Design Examples***
- Conclusion

VLSI Signal Processing Group
Chein-Wei Jen cwjen@ee.nctu.edu.tw
Department of Electronics Engineering, National Chiao Tung University

Example - Motion JPEG Video Encoder



VLSI Signal Processing Group
Chein-Wei Jen cwjen@ee.nctu.edu.tw
Department of Electronics Engineering, National Chiao Tung University

Performance Improvement of the JPEG Encoder

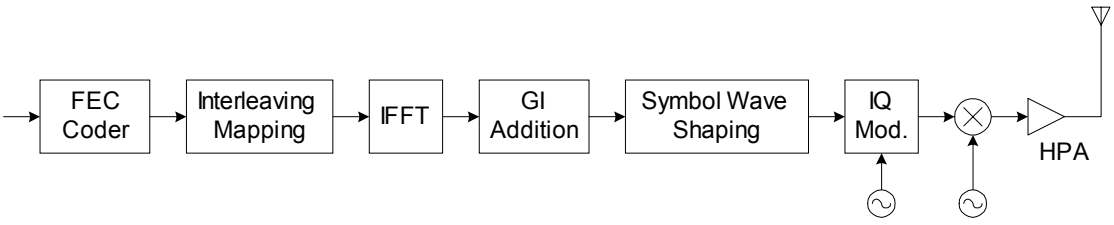
	DCT Kernel		320*240 Frame
Software alone	5,595 Cycles	111.9 us	152.928 ms
with Accelerators	246 Cycles	4.92 us	24.552 ms

- The host micro-controller is 50-MHz ARM7TDMI.
- The data-driven accelerator is composed of 4 MACs.
- 8-by-8 DCT, quantization specified in JPEG standard, run-length coding, and Huffman coding are performed on the 320*240 frame.
- An ideal memory subsystem (no memory stall) is assumed for simplicity.

VLSI Signal Processing Group
Chein-Wei Jen cwjen@ee.nctu.edu.tw
Department of Electronics Engineering, National Chiao Tung University

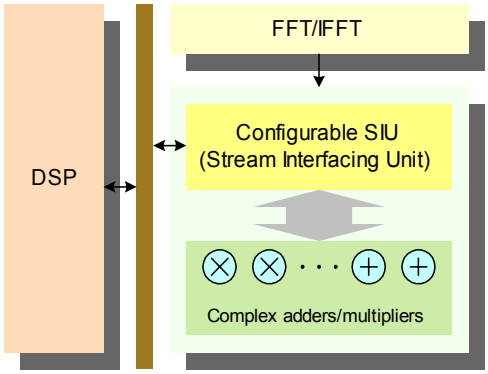
Another Example - OFDM-based Transceiver

- Transmitter



- Used in IEEE 802.11a, Hiperlan, IEEE 802.16, DVB, etc
- IFFT/FFT is the key function module

802.11a, 802.16 Specification



- Configurable FFT size
  64~8K points, 16-bit
- Latency < 3us for 64 points
  Low power
- FFT dataflow » SIU
  easy interface to DSP processor

VLSI Signal Processing Group
Chein-Wei Jen cwjen@ee.nctu.edu.tw
Department of Electronics Engineering, National Chiao Tung University

Outline

- Introduction
- Coprocessing Datapath
- CASCADE Design Environment
- Design Examples
- ***Conclusion***

Department of Electronics Engineering, National Chiao Tung University
VLSI Signal Processing Group
Chein-Wei Jen cwjen@ee.nctu.edu.tw

Conclusion

- We proposed a coprocessing datapath generation applied on the heterogeneous architecture
  - It is ***configurable*** for various DSP applications
  - It is performance ***scalable***
- The data-driven accelerators boost the performance of low-cost micro-controllers or simple DSP processors to ***lengthen the product life span***.
- Automatic generation of the accelerators with simple software-controlled interfacing dramatically ***reduces the development time***.
- The accelerating datapaths are ***driven by host instructions*** in the software driver, which is also auto-generated, to simplify the synchronization problem.