Hardware and Applications of AsAP: An Asynchronous Array of Simple Processors

Bevan Baas, Zhiyi Yu, Michael Meeuwsen, Omar Sattari,
Ryan Apperson, Eric Work, Jeremy Webb,
Michael Lai, Daniel Gurman, Chi Chen, Jason Cheung,
Dean Truong, Tinoosh Mohsenin

VLSI Computation Lab, ECE Department
University of California, Davis, USA

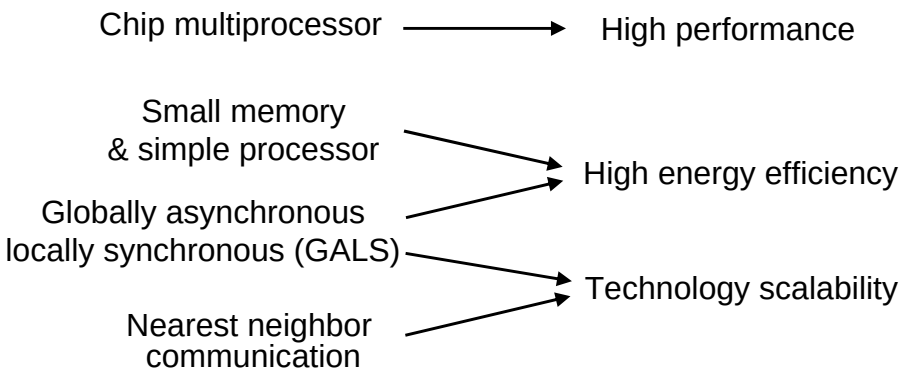
Outline

- **Motivation and key features**
- Architectural details
- The AsAP chip and results
- Programming and applications
- Conclusion

Target Applications and Objectives of the AsAP Processor

- Computationally intensive DSP and scientific apps
  - Key components in many systems
  - Require high performance
  - Limited power budgets
  - Require innovations in architecture and circuit design
- Objectives
  - High performance
  - High energy efficiency
  - Easy to program (high-level language)
  - Suitable for future fabrication technologies

Key Features of the AsAP Processor

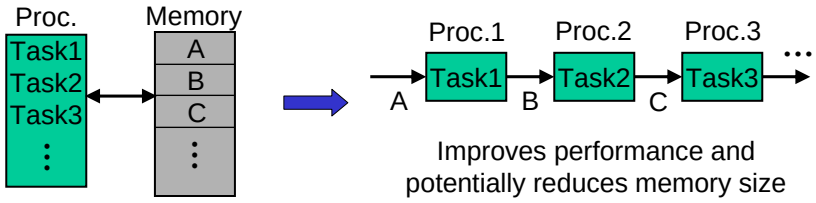


Outline

- Motivation and key features
- **Architectural details**
- The AsAP chip and results
- Programming and applications
- Conclusion

Chip Multiprocessor and Task Level Parallelism

- Increasing the clock frequency is challenging; parallelism is more promising
- Task level parallelism is well suited for and widely available in many DSP applications



Small Memory Requirements for DSP Tasks

- Memory occupies much of the area in modern processors
- The memory required for common DSP tasks is quite small
- Several hundred words of memory are sufficient for many DSP tasks

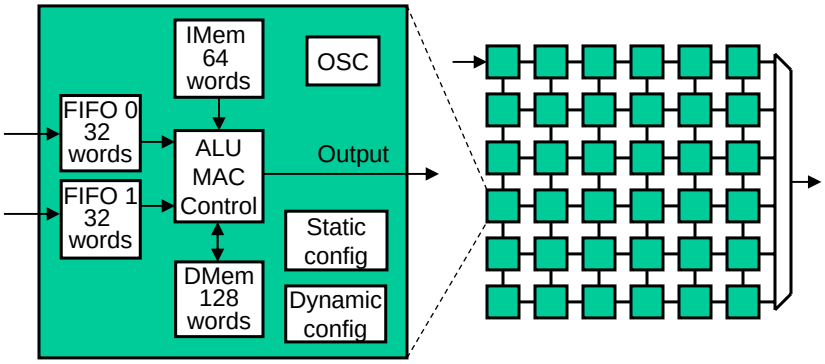
Task	IMem (words)	DMem (words)
N-pt FIR	6	2*N*
8-pt DCT	40	16
8x8 2-D DCT	154	72
Conv. coding (*k* = 7)	29	14
Huffman encode	200	330
N-pt convolution	29	2*N*
64-pt complex FFT	97	192
Bubble sort	20	1
N merge sort	50	*N*
Square root	62	15
Exponential	108	32

GALS Clocking Style and On Chip Communication

- The challenge of globally synchronous systems
  - Design difficulty due to high clock frequencies, long clock wires, and large circuit parameter variations
  - High clock power consumption and lack of flexibility to independently control clock frequencies
- The GALS clocking style addresses these challenges
- Global wires are a concern
  - Their length doesn't shrink with technology scaling—assuming the chip size remains the same
  - AsAP uses nearest neighbor communication

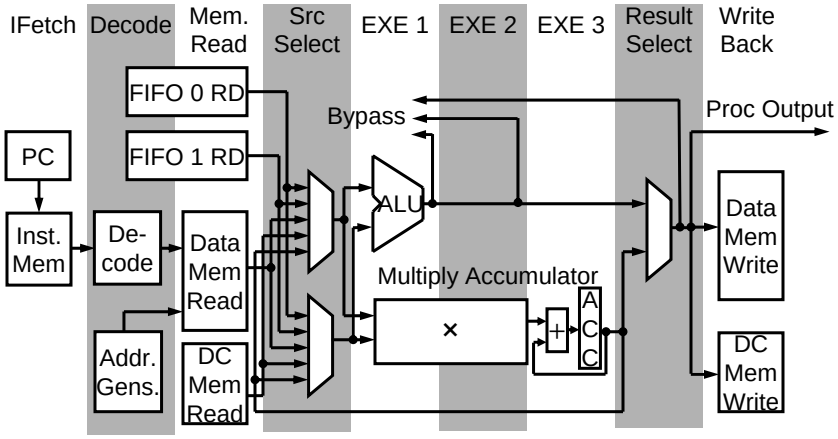
AsAP Block Diagram

- GALS array of identical processors
  - Each processor is a reduced complexity programmable DSP with small memories
  - Each processor can receive data from any two neighbors and send data to any of its four neighbors



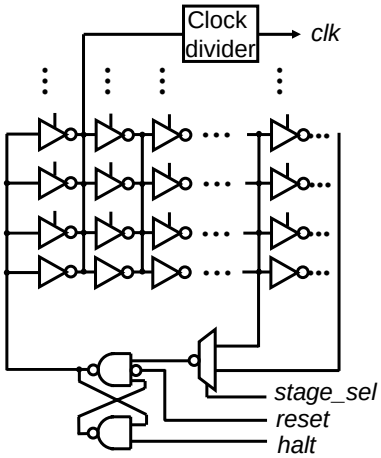
Single Processor Architecture

- 54 32-bit instructions, only Bit-Reverse is algorithm specific
- 16-bit datapath, 40-bit accumulator
- 9-stage pipeline



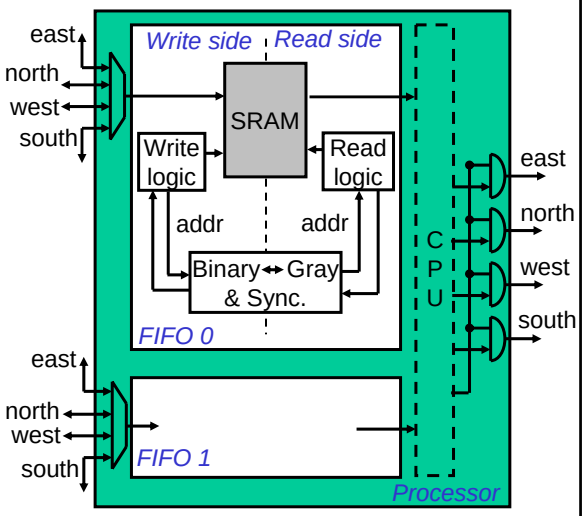
Programmable Clock Oscillator

- Configurable frequency
  - Tunable delay stages
  - 5 or 9 stage selection
  - 1 to 128 clock divider
- Circuit enables clean clock halting
- Results
  - 1.66 MHz – 702 MHz
  - Max gap: 0.08 MHz (1.66 – 500 MHz)



Inter-processor Communication

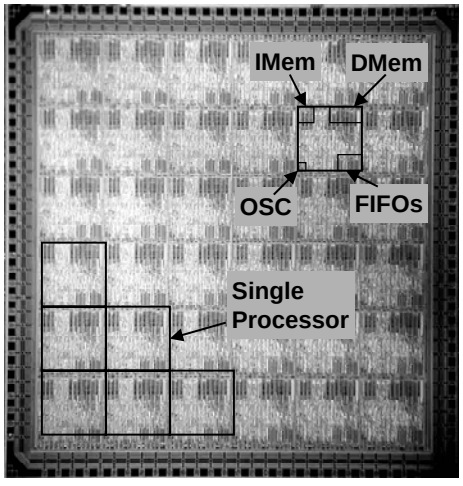
- Each processor contains two dual-clock FIFOs
- Rd/Wr in separate clock domains
  - Gray coded Rd/Wr address across clock domains
- No FIFO failures in several weeks of testing with multiple procs and no synch registers



Outline

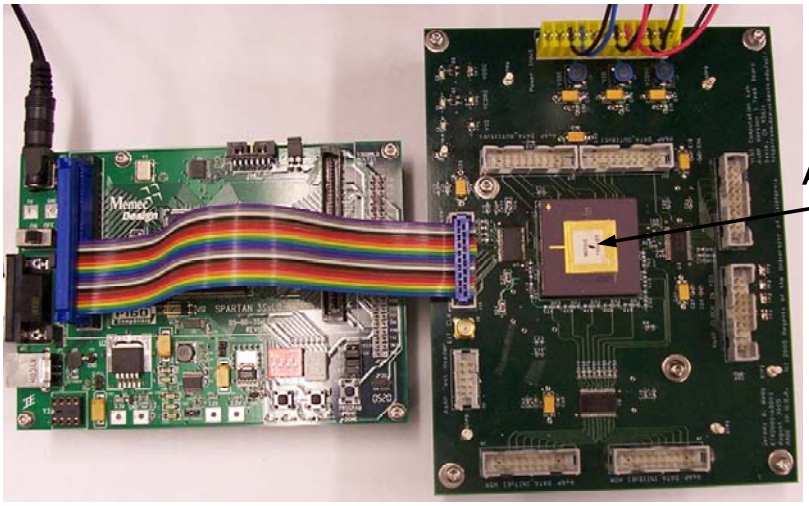
- Motivation and key features
- Architectural details
- **The AsAP chip and results**
- Programming and applications
- Conclusion

Chip Micrograph of the 6 x 6 Array

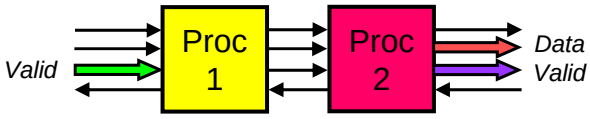


Flow:	Standard-cell based
Technology:	TSMC 0.18 µm
Transistors:	
1 Proc	230,000
Chip	8.5 million
Max speed:	**475 MHz** @ 1.8 V
Area:	
1 Proc	**0.66 mm²**
Chip	32.1 mm²
Power (1 Proc @ 1.8V, 475 MHz):	
Typical application	**32 mW**
Typical 100% active	84 mW
Power (1 Proc @ 0.9V, 116 MHz):	
Typical application	**2.4 mW**

Test Environment – PC Board and FPGA Board



Measured Clock Wavefoms



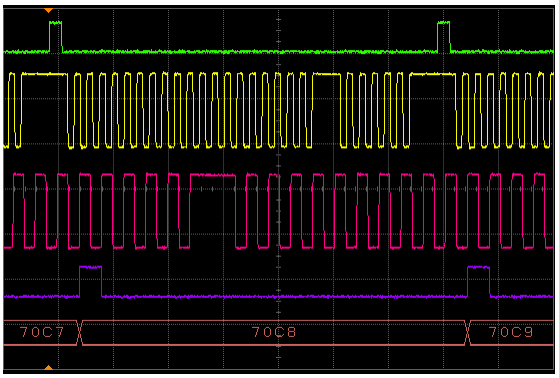
Valid for input

Clock of processor 1

Clock of processor 2

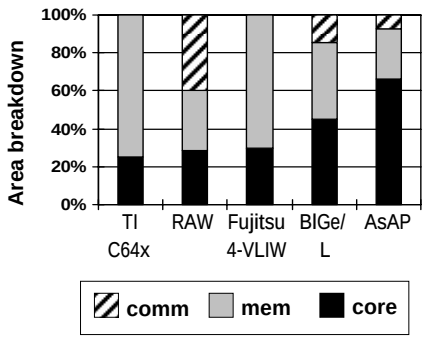
Valid for output

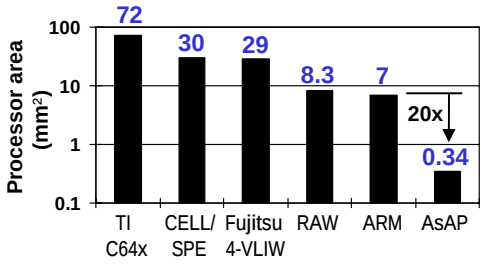
Data for output



Area Comparison

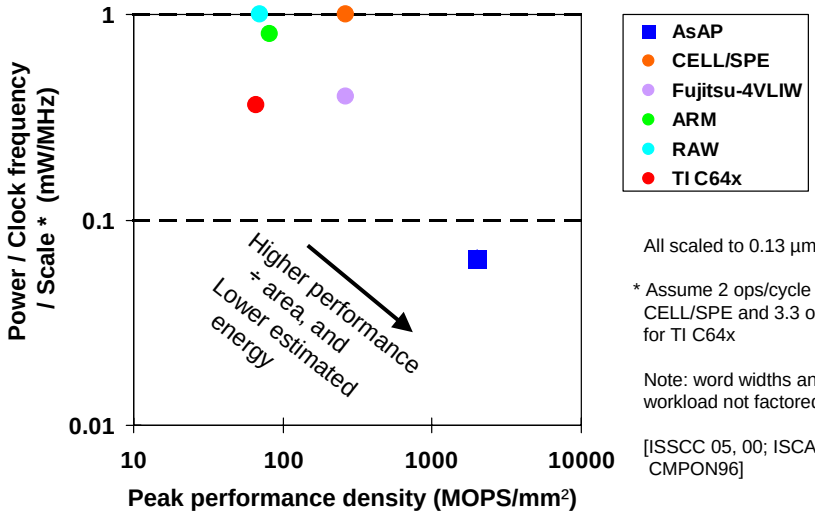
- Most of AsAP's area is for the core (66%)
- Each processor requires a very small area; more than 20x smaller than others





All scaled to 0.13 µm
[ISSCC 05, 00; ISCA04; CMPON96]

Power and Performance



All scaled to 0.13 µm

* Assume 2 ops/cycle for CELL/SPE and 3.3 ops/cycle for TI C64x

Note: word widths and workload not factored

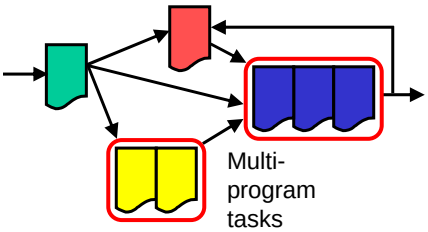
[ISSCC 05, 00; ISCA04; CMPON96]

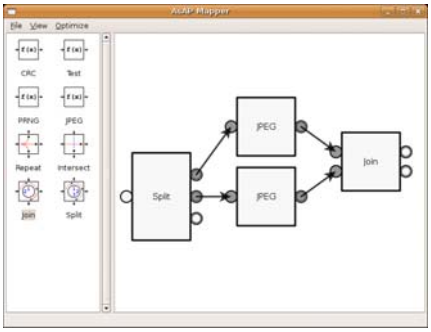
Outline

- Motivation and key features
- Architectural details
- The AsAP chip and results
- **Programming and applications**
- Conclusion

Programming

- Write C program(s) for each task
- Connect programs with an arbitrary graph
- Auto-mapping tool
  - Maps to 2-D processor array
  - Adds routing processors if needed
  - Various optimization criteria





C Programs

- Generic C with a few exceptions
  - Saturating arithmetic
  - Special variables:
    - **`Ibuf0, Ibuf1`**: input ports
    - **`Obuf`**: output port
- Stalling and async behavior completely invisible
- Example CORDIC arcsin/arccos

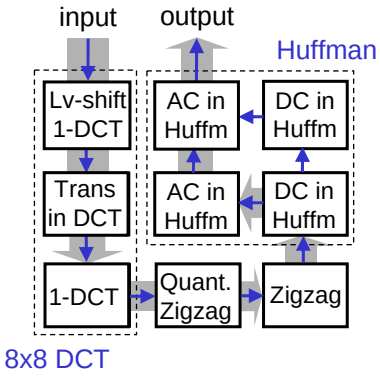
```
int atan[14];
int A = 26981, X = 16384;
int Y = 0, Z = 0;
int Xshft, Yshft, i;

Acc = A * Ibuf0;
Acc = Acc >> Ibuf0;

for (i=0; i<14; i=i+1) {
    Xshft = X >> i;
    Yshft = Y >> i;
    if (Y < Acc) {
        X = X - Yshft;
        Y = Y + Xshft;
        Z = Z - atan[i]; }
    else {
        X = X + Yshft;
        Y = Y - Xshft;
        Z = Z + atan[i]; }
    }
Obuf = 0 - Z;
Obuf = Z + 12868;
```

Benchmarks and Applications Ported to AsAP

- FIR filters (~100)
- Convolution
- Sorting (bubble, merge)
- Division
- Square root
- CORDIC sin, cos, arcsin, arccos, arctan
- Natural log
- Exponential e^x
- Pseudo random number generation
- CRC calculation
- Matrix multiplication
- Huffman encoder
- 8-point Discrete Cosine Transforms
- 8×8 2-D DCT (several)
- Fast Fourier Transforms (FFTs) of length 32-1024
- Full $k = 7$ viterbi decoder
- JPEG encoder
- Complete IEEE 802.11a/g wireless LAN baseband transmitter

JPEG Core Encoder Implementation

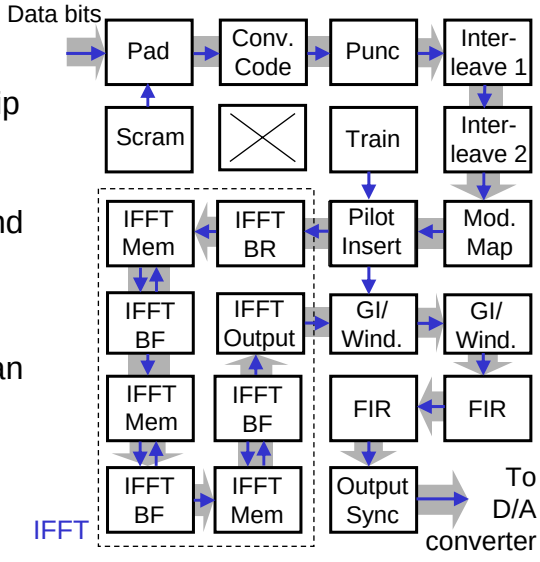
- 9 processors
- Fully functional on chip
- 224 mW @ 300 MHz
- ~1400 clock cycles for each 8x8 block
- Similar performance and ~11x lower energy dissipation than 8-way VLIW TI C62x



[MICRO 02, ISCAS 02]

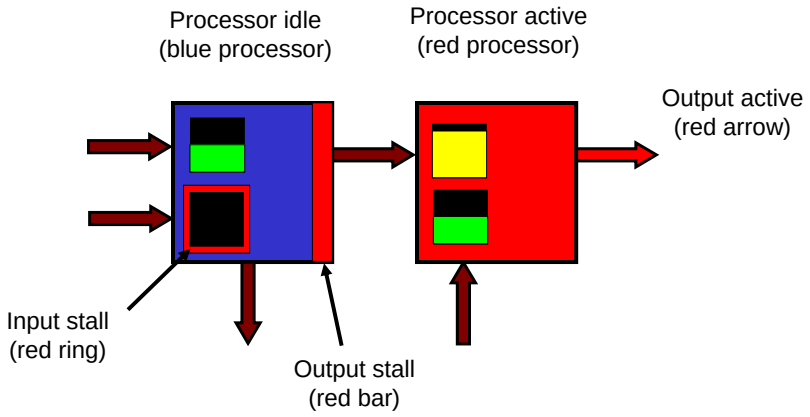
802.11a/802.11g Wireless Transmitter Implementation

- 22 processors
- Fully functional on chip
- 407 mW @ 300 MHz 30% of 54 Mb/s
- Code unscheduled and lightly optimized
- 5x - 10x performance and 35x – 75x lower energy dissipation than 8-way VLIW TI C62x

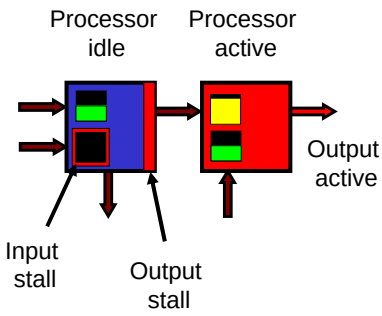


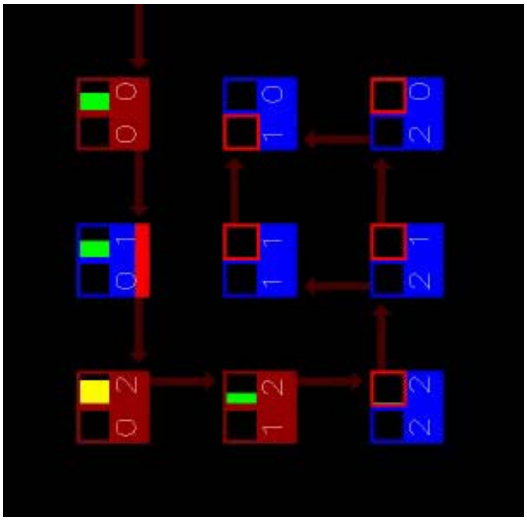
[SIPS 04; ICC 02]

Dataflow Animation

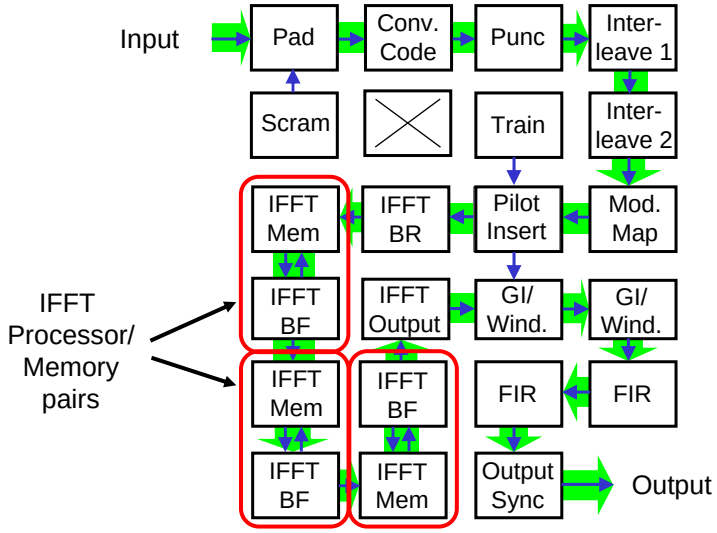


Dataflow Animation
JPEG Encoder

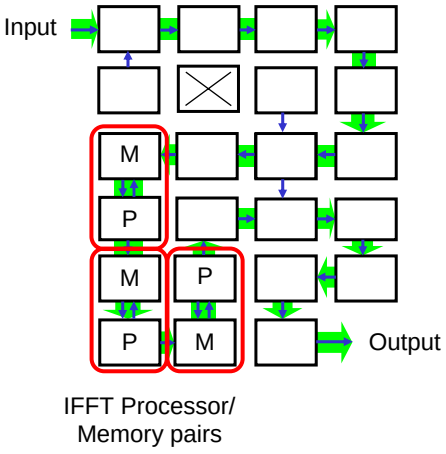


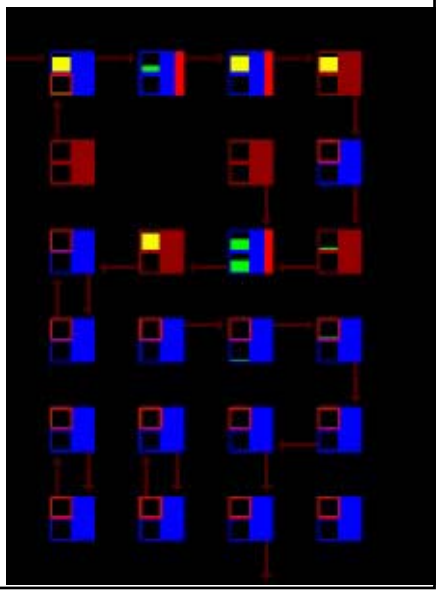


Dataflow Animation
802.11a/g Baseband Transmitter



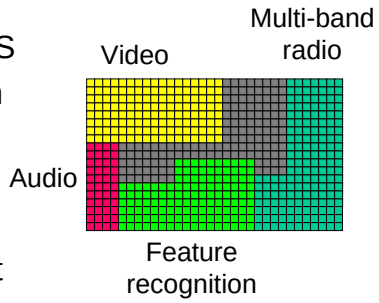
Dataflow Animation
802.11a/g Baseband Transmitter





Scaling to Hundreds and Thousands of Processors

- For the exact current design,
  - If implemented in 90 nm CMOS
  - Area slightly larger than 13mm ×13mm
  - Over 1000 processors
  - Clock rate almost 1 GHz
  - 1 TeraOp/sec peak throughput
  - Approximately 10-15 Watts application power + leakage



- Multi-application systems
- Self healing homogeneous architecture

Summary

- AsAP's key features
  - Chip multiprocessor
  - Reduced complexity processors and interconnect
    - **0.66 mm²** per processor in 0.18 µm
  - Fully independent clocking per processor
- High performance and energy efficient
  - **475 MHz** standard cell implementation in 0.18 µm
  - **32 mW** average application power at 1.8 V
  - **2.4 mW** average application power at 0.9 V, 116 MHz
- Complex multi-task applications
- C compiler and auto-mapping tool

Acknowledgments

- Funding
  - Intel Corporation
  - UC Micro
  - NSF Grant No. 0430090
  - NSF CAREER award
  - MOSIS
  - Artisan
  - UCD Faculty Research Grant
- Special Thanks
  - R. Krishnamurthy, M. Anders, S. Mathew, S. Muroor, W. Li