

TA 6.3: A 2.4GOPS Data-Driven Reconfigurable Multiprocessor IC for DSP

Alfred K. Yeung, Jan M. Rabaey

Dept. of EECS, University of California, Berkeley, CA

Rapid prototyping in hardware of high-speed real-time DSP systems, such as video codecs and HDTV, is important in shortening design time and improving design quality by providing the designer with quick feedback under realistic operating conditions. However, existing hardware prototyping platforms, often based on commercial processors or FPGAs, cannot cope with the high computation requirements of complex DSP algorithms, especially those with high sampling rate and heterogeneous data-flow patterns. The multiprocessor IC presented here is designed to handle these types of algorithms. Other approaches in this domain include the 1.5GIPS programmable VSP design based on cyclo-static scheduling and the PADDI-1 architecture with eight execution units connected by a cross-bar network [1, 2].

PADDI-2 is a MIMD multiprocessor architecture consisting of arrays of simple processing elements (PEs) connected by a reconfigurable hierarchical network (Figure 1). PEs communicate with each other exclusively using data streams and control streams in a *data-driven* manner and perform operations according to an instruction sequence stored locally. A data-flow graph representing a DSP algorithm is executed directly by mapping nodes in the graph to the PEs. The architecture uses distributed data-driven control to relieve instruction bandwidth bottlenecks, improve the ease of programming by eliminating side effects and to ensure scalability with respect to both problem size and technology. The same data-driven mechanism also handles synchronization of a large number of PEs. The basic concept is found in Reference 3.

The chip presented here contains 48 16b PEs interconnected by a 2-level high bandwidth communication network (Figure 2). Running at 50MHz, the chip provides 2.4GOPS peak. 8 of the processors also serve as I/O processors to handle off-chip communications with other PADDI-2 chips or static RAMs, and provide 800MB/s I/O bandwidth to sustain the computation performance. The 208-pin chip contains 600k transistors and measures 12x12mm² in 1 μ m 2-metal CMOS (Figure 3). Variants of this architecture that include on-chip memories are being designed.

To maximize PE usage and reduce communication cost, a flexible 2-level communication network can be configured to match the communication patterns of a wide range of algorithms (Figure 2). The level-1 communication network, consisting of 6 data buses, is responsible for communication within a cluster of 4PEs. To reduce area, the buses are laid out directly on top of the PEs as data-path feed-throughs. The level-2 communication network consists of 16 buses and handles traffic among the 12 clusters. To take advantage of locality of communications, a long bus can be broken up into several shorter ones using programmable switches increasing the effective number of buses in the network with little area penalty.

Broadcasting is supported by the network to reduce demands on point-to-point communication channels and to facilitate exploitation of data parallelism inherent in many DSP algorithms. The broadcasting and the data-driven mechanism is efficiently supported by implementing the handshake signal (HS) associated with each bus as a dynamic WIRE-AND gate (Figure 4). Data transfer is completed only if HS remains high at the end of the evaluation phase, i.e., the sender and all the receivers are ready. The long HS lines are spaced to prevent capacitive coupling from causing false discharge. To reduce area and loading on the bus,

network switches are implemented using nMOS devices only. This results in long low-to-high delay, addressed by pre-conditioning the data buses to high before driving data onto the bus. A regenerative buffer between the two levels of the interconnect network improves speed. Communications through the network uniformly take 1 clock cycle.

Because of the high data-rate and throughput requirement of the target DSP algorithms, opportunities for hardware multiplexing are limited. Therefore a simple PE architecture is used for high integration level. More complex operations are performed by configuring PEs to work in parallel, e.g., 16b x 8b multiplication can be executed at 50MHz by pipelining 4PEs.

The inset in Figure 3 shows the block diagram of a PE. It contains an 8-word by 40b program store, a 16b ALU that performs arithmetic and logic operations with special support for the modified Booth multiplication step, and one 2-word data buffer (DQi) for each of the three inputs. A conditional-select instruction speeds up maximum and minimum operations often used in signal processing. The ALU is based on a compact carry-select adder with optimized block sizes. A two-stage pipeline (fetch and execute) reduces cycle time and all instructions take one cycle to complete. The data buffer is scannable and can be configured as a data queue or a small register file. The local controller supports zero-latency branches depending on ALU status and control streams (CQi) sent by other PEs. The PE is stalled if any of the required operands are not available or if any of the output channels are blocked. To reduce network bandwidth demand, neighboring PEs can communicate without accessing the Level-1 network.

A 4b JTAG-like scan port is for downloading PE programs, configuring network switches, and presetting the contents of register files. It supports scan-testing by providing capture and single-step operation to control and observe PE inputs.

A sample of assembler code written for a PE implementing a counter is shown in Figure 5. Benchmark results for a variety of DSP algorithms in video and image processing, digital communications, and digital filtering confirm performance, efficiency, and generality of the architecture (Table 1). An implementation of an integrated Viterbi detector for (E)PRML that includes a 7-tap transversal filter and an 8-state Viterbi detector is shown in Figure 6 [4]. 34PEs are used for 12.5MB/s channel data rate, more than an order of magnitude faster than an advanced programmable DSP processor. Another example is a two-dimensional 8x8 discrete cosine transform (DCT) at 25MHz sampling rate using 50PEs and two external memories as transpose buffers. In both examples, higher performance can be achieved by using more PEs.

Acknowledgments

The PADDI project was sponsored by ARPA under the Fast System Prototyping and Infopad Projects.

References

- [1] Sluyter, R. J., et al., "A Programmable Video Signal Processor," Proc. CICC, San Diego, May, 1994.
- [2] Chen, D., J. Rabaey, "A Reconfigurable Multiprocessor IC for Rapid Prototyping of Real Time Data Paths," ISSCC Digest of Technical Papers, Feb., 1992.
- [3] Yeung, A., J. Rabaey, "A Data-Driven Architecture for Rapid Prototyping of High Throughput DSP Algorithms," IEEE VLSI Signal Processing Workshop, Oct., 1992.
- [4] Sugawara, T., et al., "Viterbi Detector including PRML and EPRML," IEEE Transactions on Magnetics, vol. 29, Nov., 1993.

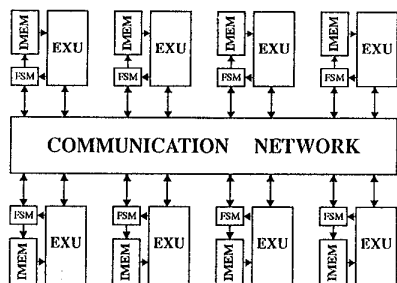


Figure 1: PADDI-2 MIMD architecture.

```

/* Sample Assembler Code: COUNTER.s */
.init q0=r, q1=r, q2=r /* All 3 data buffers are configured as register files */
.init r1 = 10 /* loop count */
.init r3 = 1 /* loop count decrement */
.init pc = START /* pc initialization */

START: r2 = sub(r1,r3), cc0 = s, next = cc0? START : LOOP;
LOOP: r2 = sub(r2,r3), cc0 = s, next = cc0? START : LOOP;
    
```

Figure 5: Sample PADDI-2 assembly code implementing a counter.

Algorithm	Performance (MHZ)	Throughput* (MOPS)	Hardware (PE)
3x3 sorting filter	50	1500	30
11-tap 8-b FIR	50	1050	56
2-D 8x8 DCT (mx'd)	25	800	50 + 2 mem
2-D 8x8 DCT (non-mx'd)	50	1600	94 + 2 mem
EPRML Viterbi detector	12.5	712	40
Motion vector estimation	22	4244	254 + 4 mem
256-pixel hidden-surface processor	12.5seg/s	32,000	640

Table 1: DSP benchmarks. *(1 16x16b multiplication counted as 1OP.)

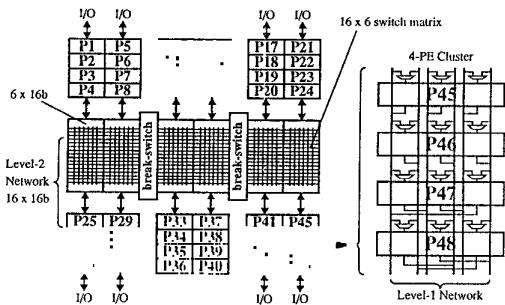


Figure 2: Block diagram of 48-PE PADDI-2 chip.

Figure 3: See page 346.

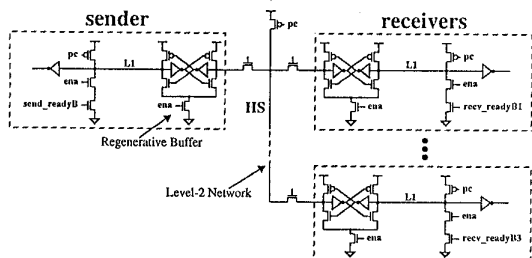


Figure 4: Dynamic WIRE-AND handshake circuit supporting broadcasting.

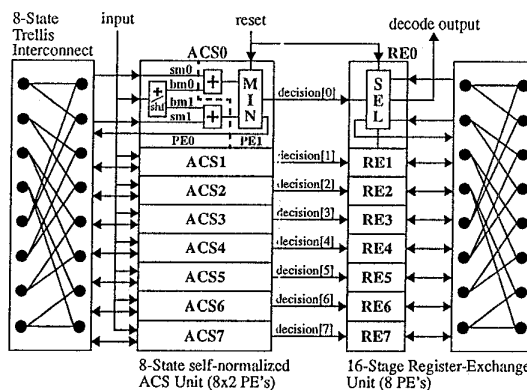
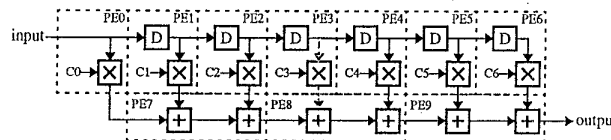


Figure 6: (a) Equalizer implemented as a 7-tap transversal filter using 10 PEs. (b) Viterbi detector using 24 PEs.

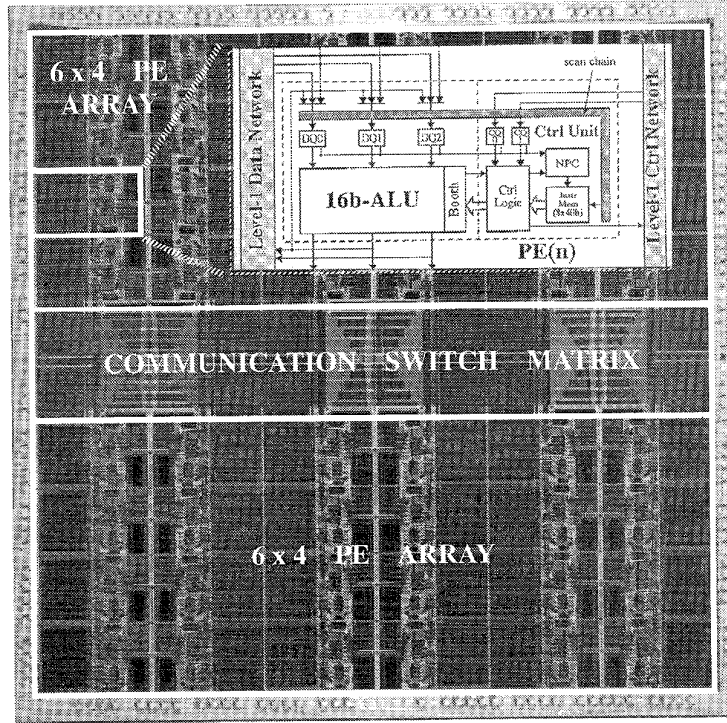


Figure 3: PADDI-2 chip micrograph.

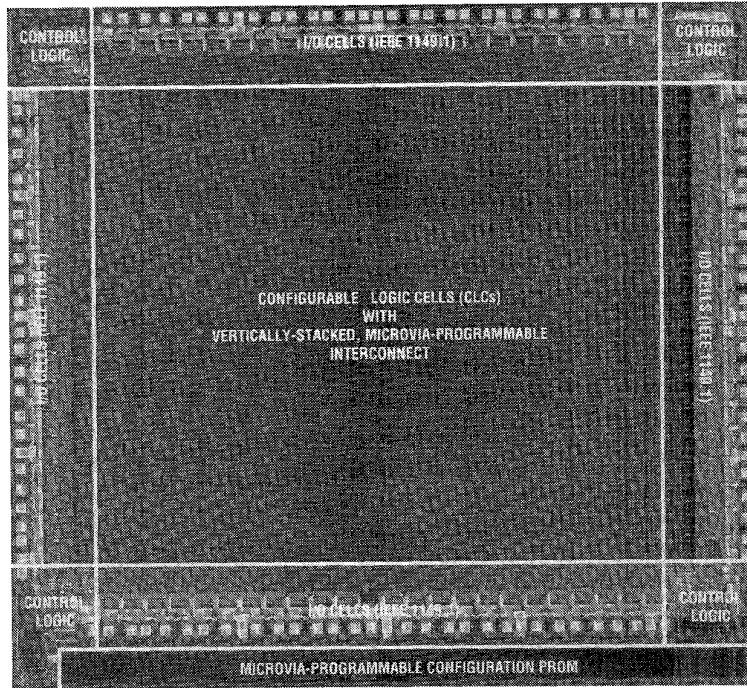


Figure 1: Sea of gates FPGA micrograph.