November 25, 2002

New NEC Array Speeds Data

NEC Introduces Its Dynamically Reconfigurable 512-Processor Array

By Max Baron

Digital media and communications are in their infancy. Most of their development and deployment roadmaps are still in the future, but they promise to become an indispensable part of everyday life. For computer architects, the new applications represent both challenges and rewards. The workloads are data intensive and require performance levels that are often impractical to implement with general-purpose processors. Challenge and opportunity are engendering specialized architectures that are competing for a chance to show their might and enjoy a slice of revenues that may rival those of the PC market. Two years ago, in Japan, NEC's research team started looking at an interesting engine that could be used in the new applications.

On October 16, 2002, at the annual **Microprocessor Forum**, Masa Motomura, an architect at NEC's System ULSI Development Division, unveiled details of the company's new massively parallel architecture, a dynamically reconfigurable processor (DRP). The new architecture can be used as a network processor or as a DSP engine in applications requiring high performance.

DRP Brings Together Three Powerful Concepts

The DRP is not the first-ever massively parallel engine, nor will it be the last, but the innovative features that set it apart really demand a second look. Three notable features stand out from the rest. To begin with, most arrays are designed as network processors or as DSP engines; the DRP can perform both functions. It can also pinch-hit as a semiefficient, but working, general-purpose processor.

Second, NEC's architects have created an architecture that can change its array configuration on a cycle-by-cycle basis, making these changes indistinguishable, timing-wise, from instructions. Most other designs have defined longer-reconfiguration delays that work best if the resulting interconnections are kept fixed for the duration of a thread.

Finally, the DRP applies a different solution to the propagation delays that must be taken into account as data moves across the chip. Where most other architectures are synchronizing units via clocked registers and processing elements (PE), the DRP can define multiple propagation paths to become one pipe stage—a small asynchronous engine walled between clocked registers to make it cooperate with other parts of the array.

PE Architecture Supports Flow-Through Data

Figure 1 shows the DRP's byte-wide processing element, which consists of a data-management unit (DMU) and an ALU designed to operate on 8-bit and 1-bit data. The DMU can execute 25 instructions that include inversion, shifting, masking, and constant generation, using 8-bit and 1-bit operands. A special command named WIRE is used to cause the DMU to pass the operand unchanged to the PE's outputs. The ALU can execute 23 arithmetic/logic instructions on 8-bit data and can use a carry propagation path to process data that is wider than 8 bits. Like the DMU, the ALU has a WIRE command.

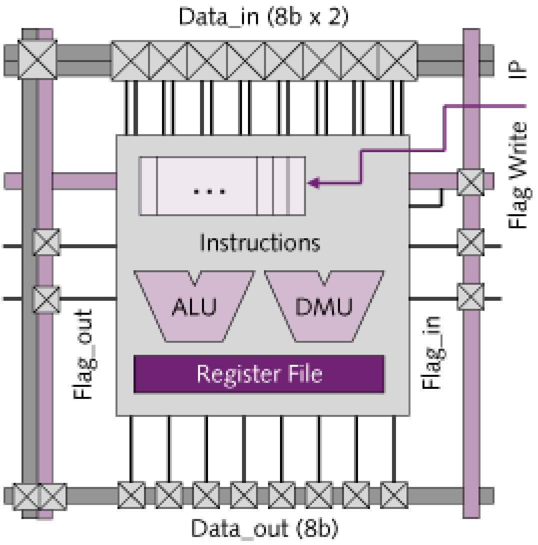

Figure 1. Simplified block diagram of NEC's processing element shows the local register file, which can hold operands and can be used as a pipe-stage register at the end of a flow-through datapath. A local stack of instructions for the DMU and ALU execution units also holds interconnect control for the PE.

Local operands are kept in the local register file. Flags that can contain arithmetic carry or condition codes are fed through for distribution to other PE units and can also be modified locally. A stack of 16 registers holds macro instructions, to be executed by the ALU and DMU, and configuration information that defines which neighboring PE units will be used as operand sources and destinations. A central finite-state machine, the state transition controller (STC), selects one of the 16 PE instruction registers via an instruction pointer. The STC can implement one-cycle configuration transitions, due to the preloaded interconnect information in the PE's local instruction registers. A collection of configuration instructions that are activated together in the same cycle forms a context, or datapath plane. A PE's instruction registers can hold up to 16 contexts. Memory access permitting, new contexts can be loaded while some of the resident ones are still active or about to become active.

Figure 2 shows DRP-1, NEC's first prototype, a chip that aims to be more independent than some of its array-processor competitors. Surrounding eight separate 64-PE tiles, each tile with its own STC and access to local memory, is a set of 32-bit multiplication accelerators and input/outputs. A total of 160KB of memory is vertically distributed, and an additional 2MB is distributed as horizontal memory. A controller interfaces the chip to SDRAM, SRAM, or CAM. CAMs can help in network processing as well as in some printer applications. A PCI bus completes the chip's interface resources. Four on-chip PLLs help distribute local clocks to the eight tiles.

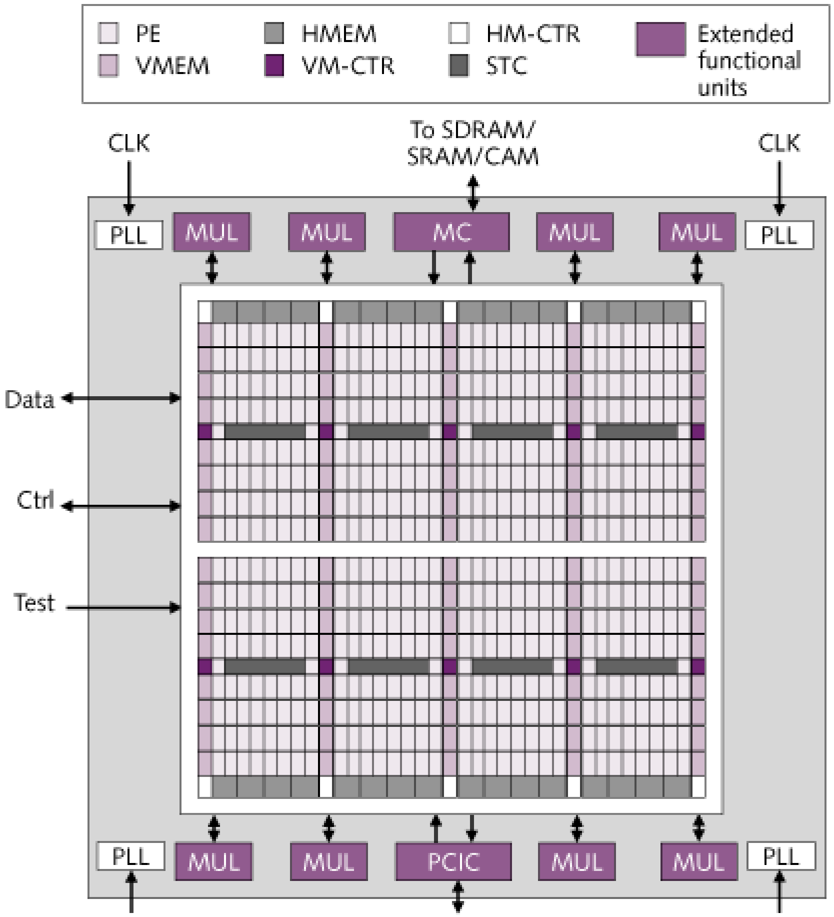

Figure 2. DRP-1 uses eight tiles, each comprising 64 processing elements managed by a separate state-transition controller. PEs have access to vertical distributed memory (VMEM). Upper and lower boundary tiles have access to the larger on-chip horizontal memory (HMEM) also.

The 512-PE DRP-1 is implemented in NEC's 0.15-micron CMOS 8-Al process and is packaged in a 696-pin TBGA package. It uses 44,000 transistors per PE, totaling 22 million transistors dedicated to PE logic. An additional 1.5MB of on-chip memory is dedicated to configuration. The DRP-1 operates at frequencies between 33MHz and 133MHz.

Programming an Application to Run on DRP

An array as complex as a DRP requires significant investment of resources in software development tools. NEC has created a C compiler that can compile source code into DRP object code. The DRP object code comprises code for all ALUs, DMUs, and configuration of PE interconnects, plus code to drive the state-transition controllers. The compilation process begins by separating the source code that will be executed by the host engine from code that must be aimed at the DRP unit(s). DRP code is input to a high-level synthesis program that generates a definition of the finite-state machine and the code for the state-transition controller. The DRP code also generates a set of interconnect definitions that form the datapath planes that will serve the workload. A mapper assigns instructions to each PE and connects PEs and memory resources, and a place and route program superimposes the mapped interconnect on top of the physical PE and memories. Finally, the code for PEs and memory interconnect is generated.

Masa Motomura of NEC unveiled details about DRP at MPF 2002. Photo by Ross Mehan.

Aside from the usual multiwindowed programmer interface, NEC's compiler offers graphic views of the scheduled dataflow graph and the scheduled state-transitions diagram. Place and route-determined connections are also displayed to help in analyzing critical-path delays. The programmer can assign a critical-path delay to be used by the high-level synthesis program. The program will divide the implementation into multiple states to fit within the critical-path-delay budget. It is expected that the visual display of information could help in speeding up place-and-route work but will be of limited use in programming and debugging complex code. The DRP has been provided with internal logic to help debug programs.

The DRP in Action

System designers must be able to take advantage of the chip's most prominent features: applicability beyond digital signal processing, one-cycle datapath change, and dataflow. NEC's architects have endowed the PE with capabilities that can support general data-intensive processing, but they had to add eight 32-bit multipliers to meet DSP needs such as could be encountered in high-end image processing. NEC's compiler provides a seamless environment for writing code aimed at PEs and multipliers.

The DRP is one of the few architectures that has brought PE program and interconnect control inside the PE itself. The DRP can switch instructions and interconnections faster than PACT or PipeRench can, but it must pay for this speed by being able to store only 16 words at a time. The programmer must ensure continuity for applications that require more than 16 configurations and instructions. If one or more configurations are long enough in execution, instruction data in some registers can be replaced by new instructions. Other opportunities for loading registers can be found during moves of data blocks and during periods of inactivity of one or more tiles.

DRP data paths' lengths and time delays may vary from one tile to another, because of the dataflow nature of the machine. For best performance, tiles can be clocked to accommodate various pipe-stage lengths and to synchronize them with one another. Clock synchronizations are also required to match internal memories to tiles and to external memory and I/O. Clock frequencies must take into consideration the dependency of flow-through delays on process, temperature, and voltage.

DRP's flow-through architecture, owing to its reduced number of clocked registers, promises higher performance and lower power consumption, and NEC will be able to make good use of it in multimedia and communications infrastructure applications.

Price & Availability

NEC's DRP is in its prototype stage. For more information, please contact support@drp.jp.nec.com.