DAP Overview

- Prototyped in 1976 by International Computers Limited (ICL), deliveries started in 1980
- Matrix of PEs
  - 32x32 for DAP 500, 64x64 for DAP 600
  - Connection to 4 nearest neighbors (w/ wrap-around), plus column & row buses
  - Each PE is one bit wide, and has 32Kb–1Mb of local memory
- DAP system = host + MCU + PE array
  - Host (Sun or VAX) interacts with user
    - Host is used for program development and debugging, and for loading and initiating DAP programs
  - Master control unit (MCU) runs main program (Fortran-Plus, APAL), broadcasts some instructions to PE array

DAP MCU and HCU

- MCU (Master Control Unit)
  - 32-bit 10 MHz CPU w/ registers, instruction counter, arithmetic unit, etc.
  - Executes scalar instructions, broadcasts others to PE array
- HCU (Host Connection Unit)
  - Gateway between DAP and host
  - Motorola 68020, SCSI port, VME interface, two RS232 serial ports
  - Provides memory boundary protection, has EPROM for code storage, 1MB RAM for data and program storage
  - Data transfers are memory-memory transfers across VME bus
  - Provides medium-speed I/O plus fast data channels (e.g.,to high-resolution color display)

DAP Processing Element

- 3 1-bit registers
  - Q = accumulator, C = carry, A = activity control (can inhibit memory writes in certain instructions)
  - All bits of a register over all PEs is called a “register plane” (32x32 or 64x64 bits)
- Adder
  - Two inputs connect to Q and C registers
  - Third input connects to multiplexor, from PE memory, output of Q or A registers, carry output from neighboring PEs, or data broadcast from MCU
    - A register also get input from this mux
    - Mux output can also be inverted
  - PE outputs (adder and mux) can be stored in memory, under control of A reg
  - D reg for asynchronous I/O, S ref for instructs that both read & write to memory

PE Memory and MCU

- PE Memory
  - Each PE has between 32 Kb and 1 Mb
  - Vector (horizontal) mode: successive bits of a word are mapped onto successive bits of a single row of a store plane
  - Matrix (vertical) mode: successive bits… onto the same bit position in successive store planes
- MCU functionality
  - Instruction fetch, decoding, and address generation
  - Executes scalar instructions and broadcasts instruction streams to PEs
  - Transmits data between PE array memory and MCU registers
  - Transmits data between DAP and host file system or peripherals

MCU

- Code store (memory)
  - 32 bit instructions, between 128 K words and 1 M words
- 32-bit general-purpose registers
  - M0 – M13: general purpose, operated on by arithmetic and logical operations, can be transferred to and from memory array
  - M1 — M7 can be used as “modifiers” for addresses and values
- Machine states
  - Non-privileged, interruptible (user mode)
  - Privileged, interruptible
  - Privileged, non-interruptible
- Datum / limit regs. for address checking

MCU Instructions

- Addresses
  - A 32-bit word, within a row or column, within a store plane
- “DO” instruction
  - No hardware overhead for these loops
  - HW support allows instructions inside the loops to access, in successive iterations, successive bit planes, rows, columns,or words of memory
- Nearest neighbor
  - Specify direction in instruction for shifts
  - For vector adds, specify whether rows or columns are being added, which direction to send carry bit
  - Specify behavior at edge of operation

Gamma II^{Plus}

- Fourth-generation DAP, produced by Cambridge Parallel Processing
- Gamma II^{Plus} 1000 = 32x32
  Gamma II^{Plus} 4000 = 64x64
- PE memory: 128Kb–1Mb
- PE also contains an 8-bit processor
  - 32 bytes of internal memory
  - D register to transfer data to/from array memory (1-bit data path) and to/from internal memory (8-bit data path)
  - A register, similar to on 1-bit processor
  - Q register, like accumulator, 32 bits wide (any one of which can be selected as an operand), can also be shifted
  - ALU to provide addition, subtraction, and logical operations