LECTURE 5

VECTOR Processors

The intricacies of programming most parallel machines are well beyond the tolerance, if not the capabilities of the ordinary programmer or scientist.

The parallel programming environment has for the most part demanded of its users a detailed knowledge of the machine architecture and the operating system in order to achieve significant gains in performance.

The pipelined vector processor is the first "parallel" architecture in which there has been a successful union of hardware and software to achieve supercomputer performance in an environment in which a programmer can take a somewhat machine independent view

By a vector processor, we mean a processor that can perform an operation in a one dimensional array of values in a single instruction.

The CRAY X-MP/Model 24 Architecture

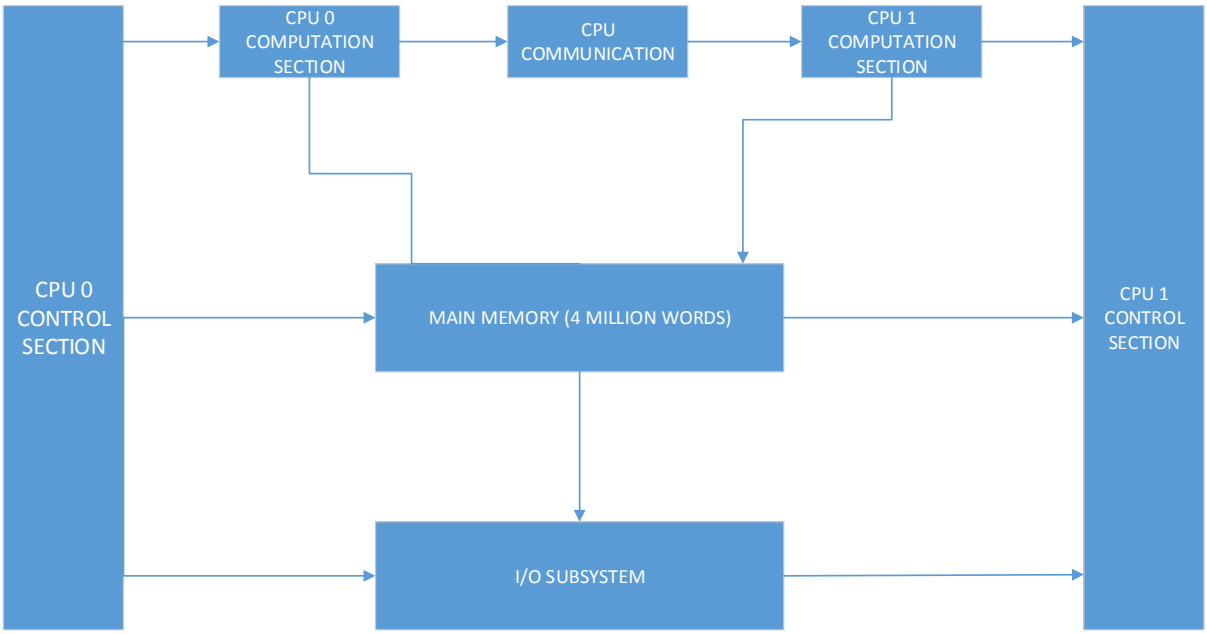

Figure 1

- The communication section includes 3 clusters of shared registers and semaphores used to arbitrate interprocessor communication and shared memory
- The main memory is shared by the two CPUs. It consists of 4 million words organized into 32 banks. Each word includes 64 data bits and 8 check bits.

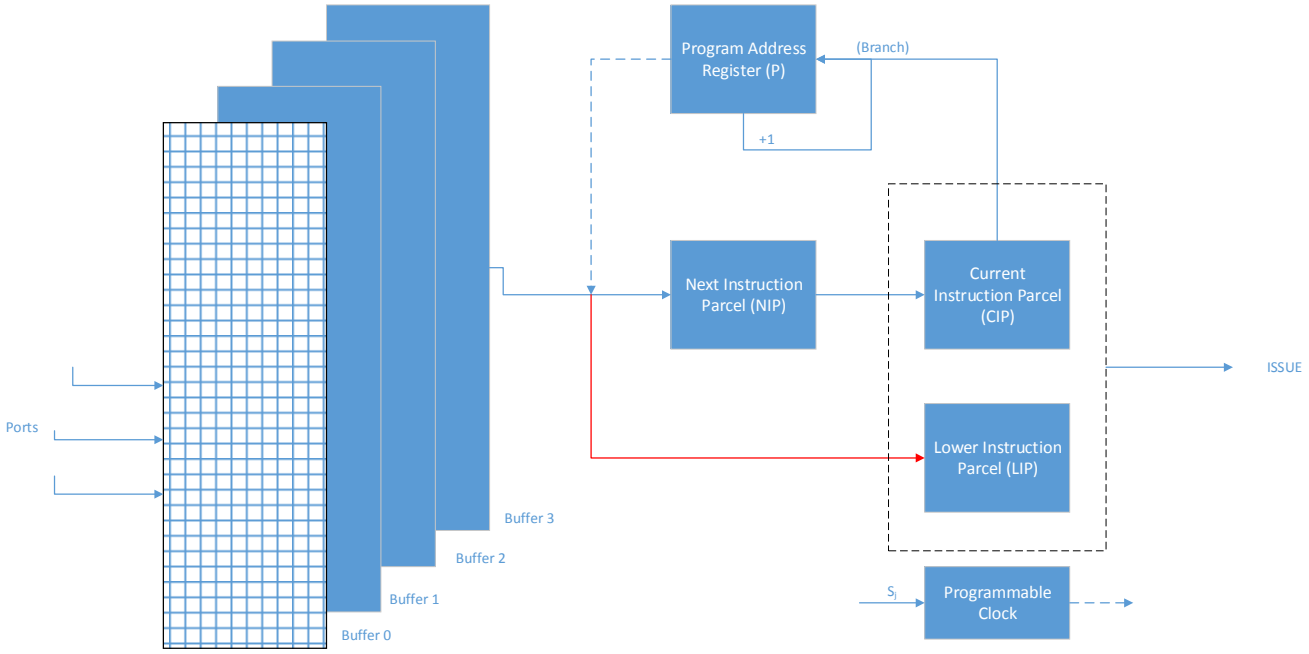

Figure 2(a) CONTROL SECTION OF CRAY X-MP
(In little grid blocks, 0a, 0b, 0c, 0d -> 1a, 1b, 1c, 1d -> …..)

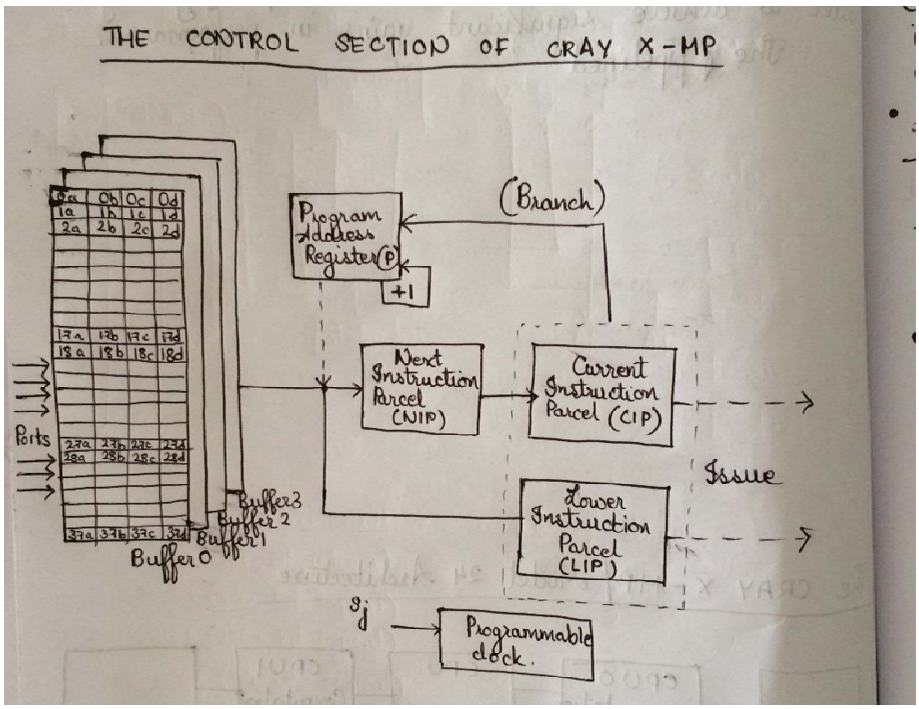

FIGURE 2(b) PHOTO FROM AKANKSHA'S NOTES. THE THINGS IN THE TINY BLOCKS WASN'T POSSIBLE TO FIT IN PREVIOUS PICTURE

Instruction Issue Phase

- The CRAY X-MP has a "LOAD and Store" architecture. Functional units require register operands. Most instructions DO NOT experience a delay due to operand fetches from memory.
- Instructions are either 16 bit (1 parcel) or 32 bit (2 parcels) long.

1-Parcel Instructions

- During the instruction fetch, the first parcel of an instruction is transferred from the instruction bugger to the NIP (Next instruction parcel register). This process takes one clock cycle.
- On the next clock cycle, the parcel is moved to the CIP (current instruction parcel register) where it is decoded. The instruction in the CIP can be issued, that is, until the processor is ready to execute it.
- If the parcel instruction will need only one clock cycle in the CIP, unless there is conflict because a previously issued instruction is using resources which will be required during the execution of the current instruction. This situation is called a hold condition.

Most of the CRAY X-MP instructions are one parcel long. Under optimal conditions, each 1-parcel long. Under optimal conditions, each 1-parcel instruction will spend one clock cycle in the NIP and one clock cycle in the CIP. These operations can overlap, so that the next instruction can be brought to the NIP on the same cycle as the previous instruction is brought to the CIP. This works as long as the next instruction is in the current instruction buffer and there are no branches. Under these optimal circumstances, an instruction issues on each clock cycle.

Branches: When a branch instruction reaches the CIP, it stays there until the instruction completes. No succeeding instructions can be issued until the branch is determined. Thus, branch instructions spend their entire instruction cycle in the issue phase and no time in the execution phase.

Instruction Issue Phase

2-Parcel Instructions:

Two-parcel instructions spend a minimum of two clock cycles in the CIP. In the cycle in which the first parcel moves from the NIP to the CIP, the second parcel is transferred from the instruction buffer to the instruction buffer to the LIP. The NIP is then filled with a zero (no operation) so that the processor does not issue an instruction on the next cycle.

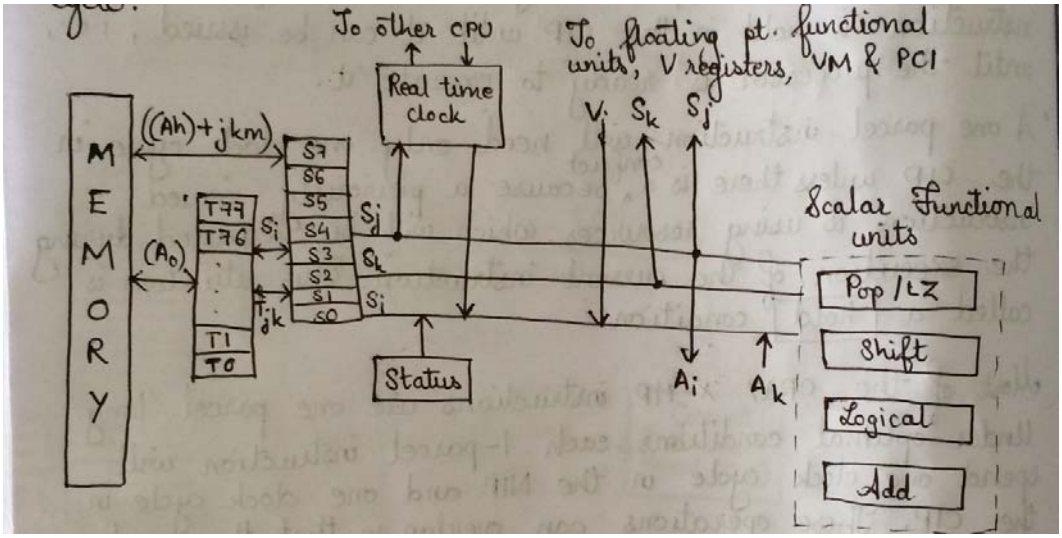

Figure 3

Scalar Functional Unit

- SCALAR ADD (64-bit Integer) ---------- 3
- SCALAR SHIFT (64- bit logical) ---------- 2
- SCALAR SHIFT (128- bit logical) ---------- 3
- SCALAR LOGICAL (64- bit logical) ---------- 1
- SCALAR POP/PARITY (population or parity) -- 4
- SCALAR POP/PARITY (leading zero count) ---- 3

(NR: The scalar floating point operations are performed in the vector functional unit.)

Observe: Scalar OUTPUT operands are RESERVED

Example:

Line	Instruction	
1	S1	S2+S3
2	S1	S2+S3
3	S2	S1+S7
4	S1	S4+S5

Instruction	1	2	3	4	5	6	7	8	9	10	11
1	I	E	E	E							
2				I	E	E	E				
3							I	E	E	E	
4								I	E	E	E

Instruction 2 cannot issue until instruction j, complete because S1 has already been reserved by instruction 1.

Instructions 3 cannot issue until instruction 2 completes because S1 has been reserved.

However, instruction 4 can issue on the clock cycle after instruction 3 issues because S1 is NOT RESERVED by instruction 3