6. Other Modifiable Models

The first published parallel model introduced and compared in power to space bounded machines was the vector machine of Pratt and Stockmeyer [PS]. A vector machine is like a random access machine, except there are two distinct kinds of registers: index registers and vector registers. Addition, subtraction, and comparison operations can be applied to both kinds of registers, and both can be accessed via index registers. In addition, bitwise Boolean operations can be applied to vector registers, and vector registers can be shifted by an amount specified by an index register. These shift operations allow the vectors to grow in length exponentially in the computation time, and hence the bitwise vector operations represent a high degree of parallelism.

Pratt and Stockmeyer prove that vector machine time $(S) \subseteq \text{DSPACE}(S^2)$ and $\text{NSPACE}(S) \subseteq$ vector machine time (S^2), for suitable $S(n) \geqslant \log n$. These inclusions are weaker than either 1.1 or 1.2. It seems that vector machines have some very powerful operations, such as the shift, which preclude linear space simulation of time. On the other hand, they are apparently not powerful enough to allow a linear time simulation of space.

This aesthetic defect is balanced by other considerations. The model is a pleasant one, and is an extension of actual computer designs. Enough examples of vector machine algorithms are given in [PS] to indicate the machine's suitability for the programming of parallel algorithms. Simon [S4] proved the surprising result that the power of vector machines is only increased by application of a polynomial when no distinction is made between index registers and vector registers, so that a vector register can be shifted by an amount specified by another vector register.

The MRAM's and CRAM's of Hartmanis and Simon [HS] are similar to vector machines, except they have only one type of register, and perform multiplication (or concatenation) instead of shifting. The time space simulation results are similar to those for vector machines.

A number of other variations of parallel random access machines have been introduced. One example is Goldschlager's SIMDAG [G1], which stands for single instruction stream, multiple data stream, global memory. This consists of a control processor (CPU) and an infinite sequence PPU_0, $\text{PPU}_1, \ldots$ of parallel processors, each connected to an infinite random access global memory. In addition, each parallel processor has a local

infinite random access memory. The program is executed by the CPU, which can broadcast instructions to the active PPU's. Each instruction broadcast is executed by the first k PPU_i's, where k is stored in some location of global memory. Each PPU_i executes the same instruction, but the memory locations accessed can be indexed by the subscript i, and so can be different for different PPU_i's. The simulations proved for SIMDAG's are a little stronger than 1.2; namely, SIMDAGTIME (S) $\subseteq$ DSPACE (S^2), and NSPACE (S) $\subseteq$ SIMDAGTIME (S). The reason that nondeterministic space S instead of just deterministic space S can be simulated in time O (S) is apparently because of a powerful SIMDAG instruction which allows any number of PPU's to store into memory at once. If two or more try to store into the same location, the lowest numbered processor succeeds. This gives the effect of a huge fan-in being executed in one step.

The P-RAM of Fortune and Wyllie [FW] is similar to the SIMDAG, except different parallel processors can be executing different parts of their program at once, so it is "multiple instruction stream". Also, there is no instruction comparable to the SIMDAG's instruction which allows a potentially unbounded number of processors to try to store in a given location at once. Wyllie shows in [W1] that the multiple instruction stream gives only a constant factor time advantage over SIMDAG's. On the other hand, the unbounded fan-in for SIMDAG's seems to be a real advantage, since the time space simulation results for P-RAM's are those of 1.2; weaker than for SIMDAG's.

The PRAM's of [SS] have no global memory, but a given processor can initiate offspring processors. The time space simulation results in [SS] are weaker than either 1.1 or 1.2.

In conclusion, all the parallel models in this section have powerful instructions which cannot be considered primitive.

7. Simultaneous Resource Bounds

In section 2 we indicated that sequential time is roughly equivalent to uniform circuit size, and sequential space is roughly equivalent to uniform circuit depth. A natural question to ask is whether simultaneous time and space bounds are roughly equivalent to simultaneous uniform size and depth bounds. To be more specific, the well known class P can be characterized as either DTIME ($n^{O(1)}$) or as USIZE ($n^{O(1)}$), and "polylog space"