Machine Specification

Physical Specifications

The machine consists of the following components:

- An infinite supply of 32-bit words. The bits are numbered from 0 (least significant) to 31 (most significant). When a word is interpreted as a number, it is an unsigned 32-bit number.
- Eight general-purpose 32-bit registers.
- A collection of arrays of words, each referenced by a distinct 32-bit identifier. One distinguished array is referenced by 0 and stores the *program*. This array will be referred to as the '0' array.
- A console capable of displaying a single ASCII character and performing input and output of unsigned 8-bit characters.

Behavior

The machine is initialized with a '0' array whose contents are read from a program file. All registers are initialized to zero. The program counter points to the first word of the '0' array, which has offset zero.

When the program file is read, each consecutively read four bytes A, B, C, D should be interpreted with A as the most significant byte, etc. (The program file size as reported by Unix ls will always be divisible by 4.)

In each cycle of the machine, an instruction is fetched from the word the program counter points to. Except for the "load program" instruction, after the instruction has been executed, the program counter advances to the next word, if any.

Instructions

There are 14 instructions. The opcode is given by the bits 28:31.

Standard Instructions

Standard instructions use three registers, A, B, and C. Each register is described by a three-bit segment of the instruction. Register A is given by bits 6:8, B is given by bits 3:5, and C is given by bits 0:2.

Opcode 0. Conditional Move. The register A receives the value in register B, unless the register C contains 0.

Opcode 1. Array Index. The register A receives the value stored at offset in register C in the array identified by B.

Opcode 2. Array Update. The array identified by A is updated at the offset in register B to store the value in register C.

Opcode 3. Addition. The register A receives the value in register B plus the value in register C, modulo 2^{32}.

Opcode 4. Multiplication. The register A receives the value in register B times the value in register C, modulo 2^{32}.

Opcode 5. Division. The register A receives the value in register B divided by the value in register C, where each quantity is treated as an unsigned 32-bit number.

Opcode 6. Nand. Each bit in the register A receives the 1 bit if either register B or register C has a 0 bit in that position. Otherwise the bit in register A receives the 0 bit.

Other Instructions

The following instructions ignore some or all of the A, B and C registers.

Opcode 7. Halt. The machine stops computation.

Opcode 8. Allocation. A new array is created; the value in the register C gives the number of words in the new array. This new array is zero-initialized. A bit pattern not consisting of exclusively the 0 bit, and that identifies no other active allocated array, is placed in the B register, and it identifies the new array.

Opcode 9. Deallocation. The array identified by the register C is deallocated (freed). Future allocations may then reuse that identifier.

Opcode 10. Output. The value in the register C is displayed on the console. Only values in the range 0–255 are allowed.

Opcode 11. Input. The machine waits for input on the console. When input arrives, the register C is loaded with the input, which must be in the range 0–255. If the end of input has been signaled, then the register C is filled with all 1's.

Opcode 12. Load Program. The array identified by the B register is duplicated and the duplicate replaces the '0' array, regardless of size. The program counter is updated to indicate the word of this array that is described by the offset given in C, where the value 0 denotes the first word, 1 the second, etc.

Special Instruction

One special instruction does not describe registers in the same way. Instead the three bits immediately below the four opcode bits describe a single register A. The remaining 25 bits indicate a value, which is loaded into the register A.

Opcode 13. Load Immediate The value in bits 0:24 is loaded into the register A (given by bits 25:27).

Exceptions

Certain "impossible behaviors" are not implemented in the machine. An exhaustive list of these Exceptions is given below. The machine must not Fail under any other circumstances.

If at the beginning of a cycle, the program counter does not point to a word that describes a valid instruction, then the machine will Fail.

If the program attempts to index or update an array that is not active, because it has not been allocated or it has been deallocated, or if the offset supplied for the access lies outside the array's capacity, then the machine will Fail.

If the program attempts to deallocate the '0' array, or to deallocate an array that is not active, then the machine will Fail.

If the program attempts to divide by 0, then the machine will Fail.

If the program attempts to load a program from an array that is not active, then the machine will Fail.

If the program attempts to output a value that is larger than 255, the machine will Fail.

If at the beginning of a cycle the program counter points outside the capacity of the '0' array, the machine will Fail.