The State Machine

This drives the rest of the logic through
the steps required for each instruction.
There are 16 states.
We always start in state 0.
Which of the other states we vist
(if any) depends on the instruction
and its parameters.
We usually dwell in a state only for
a single cycle but for some instructions
(shift, mul, div, sqrt) we may
take many cycles in the approriate
state. For these we use the H
module to count out time.

State	Description
0	This is the first state for the execution of all instructions. It suffices for the simpler register to register ALU type operations. For more complex operations we progress to other states.
1	We use this state to load the MSB of an immediate 16 bit operand.
2	A pipeline stall for calculation. (Many for shift).
3	This provides a cycle for calculation for some instructions. It is the last state for all multi-cycle instructions apart from load/store. As the last state it is used for prefetching the next instruction as well as finishing the current instruction.
4	Used to load one of the bytes when loading a 16 bit operand from memory.
5	Used to load a byte when loading either an 8 or a 16 bit operand from memory.
6	Used to store one of the bytes when storing a 16 bit operand to memory.
7	Used to store a byte when storing either an 8 or a 16 bit operand to memory.
8	Used to POP PS for RETI
9	Used to PUSH PS for TRAP
10	Most of sqrt occurs in this state.
11	Most of division occurs in this state.
12	Used for negative operand adjustment in DIVS
13	Most of multiplication occurs in this state.
14	Used for negative operand adjustment in MULS.
15	Stall for TRAP and JSR ABS.