module ..... ;
end module

task ..... ; } used regarding memory
end task

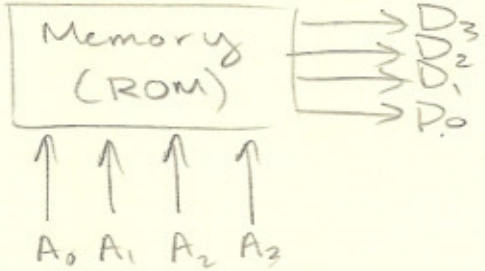
function ..... ; } $y_1 = x_1 \cdot \overline{x}_2 + \overline{x}_3$
end function

begin } All opperations in these brackets happen in sequence.
end

A bit may take on one of 4 logical variables.

1 0 X Z — X: undetermined; Z: high impedance

Consider ROM



```
input A0, A1, A2, A3;
ouput D0, D1, D2, D3;
```

varicbles must be dedared at the beginning of the module

If we had RAM memory, then we would declare the D0, D1, D2, D3. as

```
inout D0, D1, D2, D3;
```

Initization of the input takes place outside the module

```
A0 = 1;
A1 = 0;
module memory;
 :
```

note: A0 → 3 are initialized outside the module, b/c they depend not on the module, but0 rather on what is inputted to the module. D00 → D13 are declared so in the module b/c it is part of the module.

```
reg  D00 = 0;
     D01 = 1;
     D02 = 0;
     D03 = 0;
     D10 = 1;
     D11 = 0;
     D12 = 0;
     D13 = 0;
```

} also assigns varicbles