|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **address** | | | | **value** |
| **hexadecimal** | **binary inputs** | | |
| 0 | 0 | 0 | 0 | ball-x-position |
| 1 | 0 | 0 | 1 | ball-y-position |
| 2 | 0 | 1 | 0 | brick-x-position |
| 3 | 0 | 1 | 1 | brick-y-position |
|  |  |  |  |  |

4 1 0 0 score

|  |  |  |
| --- | --- | --- |
| **switch** | **state** | **effect** |
| a | Off | the address register is the B input to the ALU |
| On | a value from RAM is the B input to the ALU |
| c1 | Off | the data register is the A input to the ALU |
| On | the ALU selects 0 as its A input |
| c2 | Off | nothing |
| On | the ALU inverts its A input |
| c3 | Off | nothing |
| On | the ALU selects 0 as its B input |
| c4 | Off | nothing |
| On | the ALU inverts its B input |
| c5 | Off | the ALU performs a bit-wise AND on its inputs |
| On | the ALU performs addition on its inputs |
| c6 | Off | nothing |
| On | the ALU inverts its output |
| d1 | Off | nothing |
| On | the ALU output is written to the address register |
| d2 | Off | nothing |
| On | the ALU output is written to the data register |
| d3 | Off | nothing |
| On | the ALU output is written into RAM |

Load Value 0x0000 into the address register

Set MSB=T, a= T, c1=T, c2=T, c3=F, c4=F, c5=F, c6=F d1=F, d2=T, d3=F

Load Value 0x0002 into the address register

Set MSB=T, a= T, c1=F, c2=F, c3=F, c4=T, c5=T, c6=T, d1=F, d2=F, d3=F (TESTING IF = 0)

Load Value 0x0001 into the address register

Set MSB=T, a= T, c1=T, c2=T, c3=F, c4=F, c5=F, c6=F d1=F, d2=T, d3=F

Load Value 0x0003 into the address register

Set MSB=T, a= T, c1=F, c2=F, c3=F, c4=T, c5=T, c6=T, d1=F, d2=F, d3=F (TESTING IF = 0)

Load Value 0x0004 into the address register

Set MSB=T, a=T, c1=F, c2=F, c3=F, c4=F, c5=T, C6=F, d1=F, d2=F, d3=T