**Informe semana 2  
  
Organización de computadores**

**Practica construcción de la CPU Hack en Logisim**

**Samuel Valencia Loaiza**

**Sara Pineda**

**Facultad de Ingeniería y Ciencias Aplicadas**

**Universidad EAFIT  
Medellín, Colombia**

1. **Diseño de la ROM**

Tomando como base la información proporcionada por Nand2tetris, se decidió por utilizar la ROM que está presente como componente de Logisim, usando una configuración de “Address bit width” y “Data bit width” de 15 y 16 a cada uno para cumplir con el requerimiento de una ROM32k x 16.  
Así mismo en base a Nand2tetris se propone inicialmente para el proyecto el uso de las instrucciones de la siguiente manera, se tomará en cuenta el formato ya establecido en clase para todas las instrucciones

| **Bits** | **Campo** | **Descripción** |
| --- | --- | --- |
| 0–5 | ALU\_control | Señales para controlar la ALU (función lógica/aritmética, negaciones, etc.) |
| 6 | ALU\_srcM | 1 si el operando es M (RAM[A]), 0 si es A |
| 7 | Load\_A | Cargar valor en el registro A |
| 8 | Load\_D | Cargar valor en el registro D |
| 9 | Load\_M | Escribir en RAM[A] |
| 10-12 | Jump | Código de salto condicional |
| 13 | PC\_load | Cargar un nuevo valor en el PC (PC ← A o PC ← PC+1 según condición) |
| 14 | PC\_inc | Incrementar el PC |
| 15 | A\_mux\_sel | Selección del origen de datos para A (inmediato o salida de la ALU) |
| 16–23 | Next\_addr | Dirección de la siguiente microinstrucción si no hay salto |
| 24–31 | Jump\_addr | Dirección de la microinstrucción a saltar si se cumple la condición |

Estas instrucciones están sujetas a cambios para poder cumplir con implementar todas las instrucciones que posee el Hack CPU .