**Denotación de los componentes del PDUA**

1. Los componentes del PDUA son:

* Unidad de control
* MAR
* Banco de Registros
  + Contador de programa: PC
  + Apuntador a la pila: SP
  + Apuntador a datos: DPTR
  + Registro: A
  + Vector de interrupción: VI
  + Un registro temporal de uso interno: TEMP
  + Constante – 1: Cte1
  + Acumulador: ACC
* MDR
* ALU

El flujo que se debe implementar con los módulos del sistema en maude es el que representa el ciclo completo de “traer, decodificar y ejecutar” que esta especificado en las diapositivas dadas por el profesor Mauricio Guerrero.

Es claro que la extensión de esta especificación de cambio de estados obliga a intentar solventar los problemas sintácticos referidos al tamaño de las cadenas con módulos funcionales en la medida en que los comportamientos de los componentes sean expresables de este modo. Este fue el camino llevado a cabo con respecto a la expresión del comportamiento de la ALU, de manera que no se necesitó entrar a escribir cada una de las operaciones con todas las posibles escogencias de elementos del dominio de cada una de estas (números binarios de 8 bits).

Se requiere para las cadenas que denotan los dispositivos de alto nivel que los componentes que las conforman tengan significado para el posible usuario de la especificación. Podríamos simplemente escribir todos los componentes usando los códigos para cada una de las instrucciones directamente en binario. Si bien esto es transparente para el interpretador maude, para el especificador seria un problema entender el significado, aunque esto podría solucionarse levantando una interfaz gráfica conectada al interpretador por medio de las facilidades de conectividad de este.

Maude soporta el disparo de varios hilos de reescritura diferentes si se cumplen dos o más reglas especificadas dentro de un módulo del sistema. Esta capacidad de maude no necesitó ser usada dado que el procesador modelado no implementa en algún sentido un flujo de datos o de control concurrente, o un comportamiento que requiera escribir alguna regla con dos posibles flujos disyuntos.

1. Prototipos de cadenas
   1. Cadena PDUA

La cadena PDUA debe colapsar en si todos los componentes que hacen parte del dispositivo.

Dentro de la especificación VHDL, los dispositivos que interconectan las entidades del PDUA son los buses. Para efectos de la misma especificación VHDL, estos dispositivos no tienen más denotación que entrar a ser proyecciones de puertos escritas en la especificación de los componentes de ellos mismos. En maude entonces, la denotación de los buses vendría a ser una declaración implícita que toma estados de un componente (representado por una cadena) y por la misma semántica que tiene la regla de transición llevaría a la cadena a el contexto del siguiente estado según este haya sido determinado por la especificación VHDL.

Proponemos entonces la siguiente cadena:

<UC><Banco><ALU><MDR><MAR>

Esta cadena representa un estado legal de la máquina. Ahora, hay que entrar a decantar el contenido de cada entidad y proyectarla dentro de cada una de las cadenas dadas de manera tal que, como se dijo anteriormente, los estados se reduzcan al mínimo.

Tomemos como ejemplo el componente MAR:

MAR tiene como responsabilidad el manejo de la activación de direcciones para ubicar las palabras dentro de la memoria. Su comportamiento se reduce a cambiar el estado del bus de direcciones si y solo si