**Dispositivos de Lógica programable.Compilador Universal de Lógica Programable (CUPL).**

Para desarrollar la programación de este tipo de dispositivos , podemos usar los lenguajes

* ABEL **Advanced Boolena Equation Lenguaje**
* OPAL **Open Architecture Languaje** de National Semiconductors
* VHDL **Verilog Hardware Description Languaje**
* CUPL **Compilador Universal de Lógica Programable de** ATMEL

Usaremos este lenguaje como mediop de desarrollo y programación.La programación se efectúa mediante la creación de un archivo de texto que contiene el código para la programación del dispositivo. Este archivo tiene tres partes básicas: el encabezado, la declaración de los pines de entrada y salida las definiciones lógicas.

La asignación de pines se puede hacer de forma individual o grupal usando nombres de variables:

|  |  |
| --- | --- |
| **Sintaxis General** | **Sintaxis Abreviada** |
| Pin 1 = Nombre; | Pin [2,3] = [Nombre, Nombre2]; |
| Pin 2 = !Nombre; | Pin [2,3] = ![Nombre, Nombre2]; |
| Pin 3 = !SET; | Pin [2..3] = [Q0..3]; |

Las variables intermedias corresponden a variables asignadas a una ecuación lógica pero que no representan un *pin* en el dispositivo. Generalmente esta variables se utilizan cuando se requiere manejar varias variables de entra y salida. Las ecuaciones lógicas indican el manejo del circuito estas usan los elementos siguientes para su escritura:

|  |  |  |
| --- | --- | --- |
| **Operador** | **Función** | **Formato de CUPL** |
| & | AND | A&B |
| # | OR | A#B |
| ! | NOT | !A |
| $ | XOR | A$B |

La sintaxis general de las ecuaciones lógicas en CUPL es la siguiente:

[!] var [.ext] = exp;

Existe otra forma de expresar las funciones cuando estas usan variables consecutivas:

|  |  |
| --- | --- |
| **Forma Convencional** | **Forma Abreviada** |
| A3 & A2 & A1 & A0 | [A3, A2, A1, A0]:& |
| B3 # B2 # B1 # B0 | [B3..B0]:# |
| C3 $ C2 $ C1 $ C0 | [C3, C2, C1, C0]:$ |

Otra forma de notación es la de rango de datos donde primero se define el rango:

FIELD entrada = [A3..A0];

Field es el nombre del campo compartido del dato.

Por ejemplo si tenemos:

**salida = entrada:C # entrada:D # entrada:E # entrada:F;**

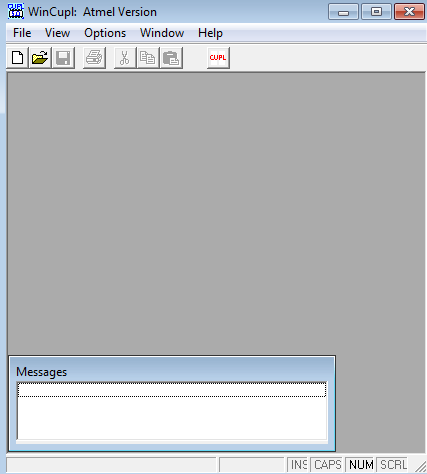
Se puede expresar de la forma siguiente:

**salida = entrada:[C..F];**

Para usar el programa una vez instalado debemos dar clic en el icono para llamar la interfase.

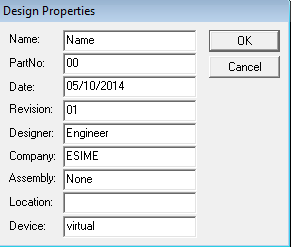


Oprimimos el Menú File

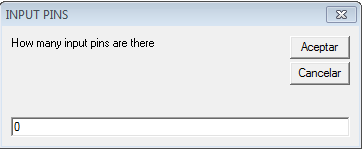


Donde dice NAME **compuertas**

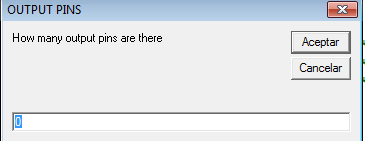
DEVICE **g16v8a**



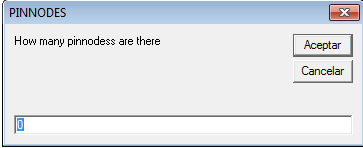
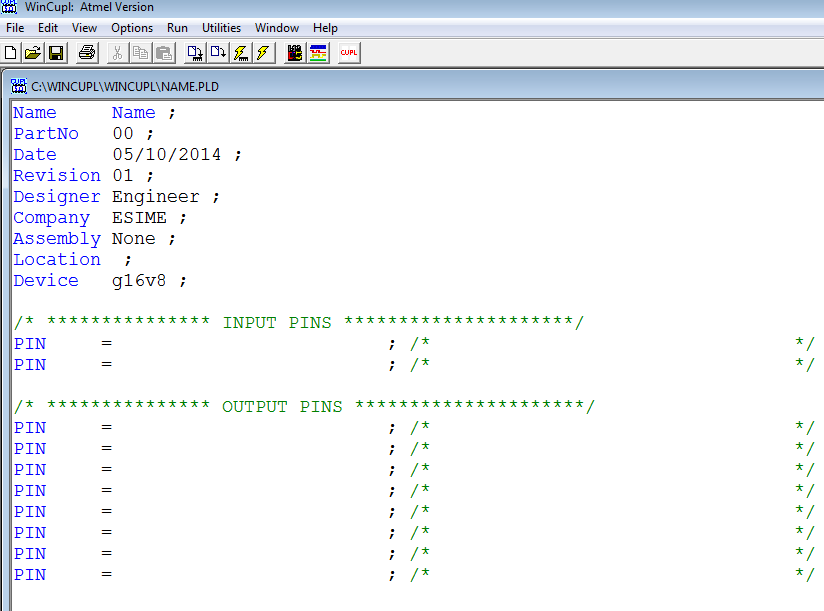
Proponemos el número de entradas **2**



Proponemos el número de salidas **8**



Como último se mencionan los elementos intermedios **0**



Dando por resultado esta pantalla

**Para este caso tenemos este ejemplo (NOTA: si copias y pegas el ejemplo cuidado ya que hay caracteres que el compilador marca como error):**

Name compuertas;

Partno CA0001;

Revision 01;

Date 5/10/14;

Designer Carlos A. Estrada;

Company ESIME;

Location None;

Assembly None;

Device g16v8a;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Como Crear un arreglo de compuertas \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

\* Entradas

\*/

Pin 1 = a;

Pin 2 = b;

/\*

\* Salidas

\*/

Pin 12 = inva;

Pin 13 = invb;

Pin 14 = and;

Pin 15 = nand;

Pin 16 = or;

Pin 17 = nor;

Pin 18 = xor;

Pin 19 = xnor;

/\*

\* Ecuaciones

\*/

inva = !a; /\* inversores \*/

invb = !b;

and = a & b; /\* and \*/

nand = !(a & b); /\* nand \*/

or = a # b; /\* or \*/

nor = !(a # b); /\* nor \*/

xor = a $ b; /\* xor \*/

xnor = !(a $ b); /\* xnor \*/