Manual del Usuario

Lenguaje: Kuac v2.0.1

Revisado el 5 de Mayo del 2015

**Introducción**

Kuac es un lenguaje basado en C para la simulación de circuitos digitales de una manera sencilla de programar y compilar, orientado a los estudiantes con nociones básicas de programación y que cuenten con niveles básicos de algebra booleana para su mejoramiento en estas dos áreas.

Kuac puede simular desde compuertas básicas como AND, OR, NOT, hasta Unidades Lógicas y Aritméticas completas de una manera sencilla con sus definiciones de parámetros y construcción de circuitos por medio de estructuras de sintaxis sencillas de recordar, las cuales serán desarrolladas a lo largo de este manual de usuario.

**Léxico**

Palabras reservadas:

|  |  |  |  |
| --- | --- | --- | --- |
| ENTITY | IS | IN | OUT |
| SIGNAL | BIT | VECTOR | TO |
| END | ARCHITECTURE | OF | BEGIN |
| IF | THEN | ELSE | CASE |
| WHEN | BREAK | OTHER |  |

Simbolos:

|  |  |  |
| --- | --- | --- |
| ( Open Paréntesis | SHL Corrimiento a la Izquierda | < Menor que |
| ) Close Paréntesis | SHR Corrimiento a la Derecha | & AND |
| : Dos Puntos | != Diferente de | | OR |
| ; Punto y Coma | = Asignacion | ! Inverso |
| == Comparación | > Mayor que | ^ XOR |
| + Suma | - Resta |  |

**Sintaxis**

La estructura base del programa consta de dos secciones la entidad y la arquitectura.

**Entidad**

La primera sección de un programa basado en Kuac, es conocida como la Entidad, es en este lugar donde se definen todas las entradas, las salidas y las señales intermedias que ayudan a la comunicación interna del circuito a simular. Este espacio es conocido formalmente como Entity y tiene la siguiente definición de sintaxis.

entity Ejemplo is(

A: in vector (3);

B: in bit;

X: out vector (4);

Y: signal vector (2);

)end P;

A es un vector de entrada, en el cual se especifica su longitud en los paréntesis siendo de 3 bits este pudiendo representar en este caso valores de 0 a 7.

B es un bit de entrada, lo cual solo le permite representar el valor de 1 o 0.

X es un vector de salida de 4 bits, representado del mismo modo que A, teniendo la posibilidad de representar números del 0 al 15.

Y es una señal interna del programa para comunicar las señales del circuito a simular, las señales intermedias no se pueden definir por el usuario más que en esta sección del programa y teniendo la posibilidad de manipularlas durante las instrucciones del mismo ya que una vez en ejecución.

Esta sección es la más crítica del programa basado en Kuac ya que durante ejecución el usuario solo alimentara al programa con las entradas digitales y el programa mostrara los resultados finales, sin permitirle al usuario el poder observar las señales intermedias definidas en esta misma sección las cuales son utilizadas para la comunicación intermedia de las entradas con las salidas durante la ejecución de la solución.

**Arquitectura**

La segunda mitad del programa es la Arquitectura, es aquí donde se describe el circuito a simular completamente basado en operaciones booleanas y/o aritméticas.

La estructura base de la arquitectura del programa es la siguiente:

architecture of Ejemplo is

begin

% % % % % % %

Definición del circuito

% % % % % % %

end Ejemplo

Para realizar la definición del circuito a simular, Kuac cuenta con operaciones booleanas y algebraicas, las cuales combinadas con las estructuras definidas a continuación se puede simular cualquier circuito digital desde operaciones básicas de compuertas múltiples, hasta complejos circuitos capaces de recrear todo el funcionamiento de un procesador integro con todos sus componentes.

Las estructuras que se pueden llegar a utilizar para realizar los diseños lógicos en Kuac son los siguientes cada uno detallado con su sintaxis correspondiente.

**IF**

Función de condición en la cual solo se realizará las operaciones de ser verdaderas, su sintaxis es la siguiente:

if(Condición) then

% % % % % %

Instrucciones Verdaderas

% % % % % %

end if;

**IF ELSE**

En caso de querer comprobar más de una posible condición como verdadera se realizara la siguiente sintaxis:

if(Condición) then

% % % % % %

Instrucciones Verdaderas

% % % % % %

else IF(Condicion) then

% % % % % %

Instrucciones Verdaderas

% % % % % %

else

% % % % % %

Instrucciones Falsas

% % % % % %

end if;

**Case**

La estructura más versátil para probar múltiples casos ligados al resultado de una sola variable, útil para probar Multiplexores y Unidades lógicas completas.

Case **Variable** is:

When 0:

% % % % % %

Instrucciones Verdaderas

% % % % % %

break;

When 1:

% % % % % %

Instrucciones Verdaderas

% % % % % %

break;

...

...

...

...

when others:

% % % % % %

Instrucciones Verdaderas

% % % % % %

break;

end case;

En este ejemplo el análisis del Case se basa en **Variable** que previamente se debió de definir con algún valor basado en operaciones booleanas y/o algebraicas y dependiendo del valor que esta tenga entrara en alguno de los casos descritos, en caso de no caer en ninguno entrara por default a la instrucción “others”.

**Operaciones Booleanas y Algebraicas**

Al ser Kuac un lenguaje para la simulación de circuitos digitales las principales operaciones a realizar serán las 4 elementales AND, OR, NOT y XOR, el lenguaje tiene una sintaxis le lectura de Izquierda a Derecha. Contando también con una jerarquía de operaciones en el siguiente Orden.

|  |  |
| --- | --- |
| Mayor Jerarquía | ( ) |
|  | SHL SHR |
|  | + - |
|  | ! |
| Menor Jerarquía | & | ^ |

Nota:

Para realizar diversas operaciones en una sola línea de comandos Kuac requiere del uso de paréntesis para limitar las operaciones, en caso contrario las realizará siempre de derecha a izquierda.

**Ejemplo básico de programa en Kuac**

entity P is(

A: in vector (2);

B: in bit;

C: in bit;

D: out vector (3);

E: signal vector (2);

) end P;

architecture of P is

begin

E=A|B;

case E is

when 0:

if(C==0) then

D=000;

else

D=101;

end if;

break;

when 1:

D=A+B&C;

break;

when 2:

if(C==1) then

D=010;

end if;

break;

when 3:

D=(A-B) SHL C;

break;

when other:

D=111;

break;

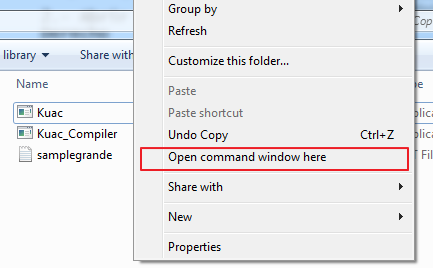
end case;

end P;

**Compilación del Código en Kuac**

1.- Ubicar el archivo con el programa realizado en la misma carpeta que el compilador Kuac\_Compiler.exe

2.- Abrir línea de comando en la carpeta con Shift + Click Derecho

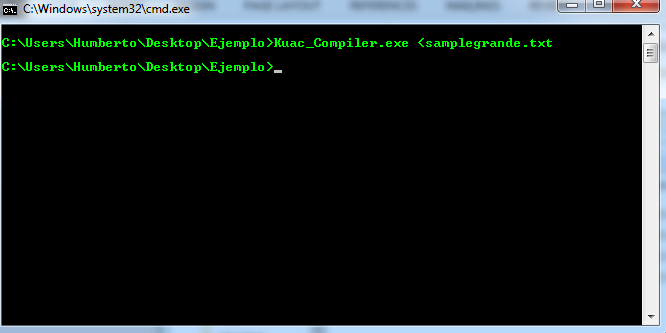


3.- Ejecutar la siguiente línea de comando

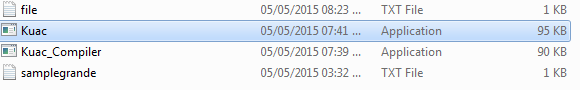
Kuac\_compiler.exe <NombreDelArchivo

En nuestro ejemplo:

Kuac\_compiler.exe <samplegrande.txt



4.- Una vez que el archivo termine de compilar se generará un archivo file.txt, el cual será utilizado por la máquina virtual para ejecutar el código programado en Kuac.



5.- Correr Kuac.exe

6.- Ingresar Valores de Entrada Según las especificaciones del programa original.

7.- El programa entrara en un Loop infinito para probar una y otra vez valores de entrada que el usuario proporcione, hasta que el mismo ya no requiera hacer un mayor número de pruebas y cierre el programa.