

Seminario de Solución de Problemas de Arquitectura de Computadoras



**Centro Universitario de Ciencias Exactas e
Ingenierías**

Proyecto Final parte 2

Profesor López Arce Delgado Jorge Ernesto

Equipo 2 Fabián Quezada López

Solórzano Maldonado André

Ramsés Aarón Cruz Gallegos

Descripción de los módulos

pc

Posee dos entradas “inputpc” de 32 bits y “clk” de 1 bit el cual es el ciclo de reloj y una salida de registro “outputpc” de 32 bits, tiene un bloque always que hace que en cada ciclo de reloj en la salida (outputpc) salga lo que esta en la entrada (inputpc).

meminstruct

Posee una entrada “inputmeminstruct” de 32 bits y una salida “outputmeminstruct” también de 32 bits, en el módulo esta una asignación la cual consiste en colocarle a la salida (outputmeminstruct) una concatenación de la entrada (inputmeminstruct) y después la entrada (inputmeminstruct) +1 y después la entrada (inputmeminstruct) +2 hasta el +3 para lograr obtener las primeras 4 instrucciones.

buffer1

Posee tres entradas “instrucción_in” de 32 bits, “pcsumain” también de 32 bits y “clk” de 1 bit como ciclo de reloj, junto con dos salidas de registro “instrucción_out” de 32 bits y “pcsumaout” también de 32 bits, tiene un bloque always que hace que cada ciclo de reloj, los valores de salida (instrucción_out y pcsumaout) van a tomar los valores de entrada (instrucción_in y pcsumain).

buffer2

Posee 15 entradas las cuales son “clk” como ciclo de reloj, “regwrite_in”, “memtoreg_in”, “memwrite_in”, “memread_in”, “branch_in”, “alusrc_in” y “regdst_in” de 1 bit cada uno, “aluop_in” de 3 bits, “instrucción_in” y “instrucción2_in” de 5 bits cada uno y por ultimo “pcsumain_in”, “data1_in”, “data2_in” y “signextender_in” de 32 bits cada uno también tiene 14 salidas de registro “regwrite_out”, “memtoreg_out”, “memwrite_out”, “memread_out”, “branch_out”, “alusrc_out” y “regdst_out” de 1 bit cada uno, “aluop_out” de 4 bits, “instrucción_out” y “instrucción2_out” de 5 bits cada uno y por ultimo “pcsumain_out”, “data1_out”, “data2_out” y “signextender_out” de 32 bits cada uno. Las entradas del BUFFER 2 son las salidas de la unidad de control, del banco de registros, sign extender y dos salidas de la instrucción de 32 bits proveniente del BUFFER 1. Al igual que el BUFFER 1, tiene un bloque always que cada ciclo de reloj hace que las salidas tomen los valores de entrada.

buffer3

Posee 11 entradas las cuales son “clk” como ciclo de reloj, “regwrite_in”, “memtoreg_in”, “memwrite_in”, “memread_in”, “branch_in” que vienen directamente del BUFFER 2, “zflag_in”, “data2_in” y “instrucción_in” de 1 bit cada uno, “branch_result_in” y “alures_in” de 32 bits junto con 10 salidas de registro “regwrite_out”, “memtoreg_out”, “memwrite_out”, “memread_out”, “branch_out”, “zflag_out”, “data2_out” y “instrucción_out” de 1 bit cada uno, “branch_result_out” y “alures_out” de 32 bits. Tiene un bloque always que hace exactamente lo mismo que los

anteriores buffers, las salidas toman los valores de entrada, 4 de las salidas se quedan en la memoria de datos las cuales son “memread”, “memwrite”, “ALURes” y “Data2” y se van 5 más al siguiente buffer.

buffer4

Posee 6 entradas las cuales son “clk” como ciclo de reloj, “regwrite_in” y “memtoreg_in” de 1 bit cada uno, “writeregister_in” de 5 bits, “memres_in” y “alures_in” de 32 bits, junto con 5 salidas de registro “regwrite_out” y “memtoreg_out” de 1 un bit, “writeregister_out” de 5 bits, “memres_out” y “alures_out” de 32 bits. Misma función que los buffers anteriores, con cada ciclo de reloj las salidas toman los valores de entrada.

extendersigno

Tiene una entrada “inputsigno” de 16 bits y una salida “outputsigno” de 32 bits. Este modulo sirve para las instrucciones inmediatas, cambia de 16 bits a 32 bits verificando el bit mas significativo, si es 1 se le agregan 16 1’s, si el bit es 0 se le agregan 16 0’s.

shiftright

Tiene una entrada “inputshiftright” de 32 bits y una salida de registro “outputshiftright” de 32 bits. Sirve para hacer el branch.

Instrucción	Tipo	Sintaxis
Add	R	Add \$rd, \$rs, \$rt
Sub	R	Sub \$rd, \$rs, \$rt
Mul	R	Mul \$rd, \$rs, \$rt
Div	R	Div \$rd, \$rs, \$rt
Or	R	Or \$rd, \$rs, \$rt
And	R	And \$rd, \$rs, \$rt
Xor	R	Xor \$rd, \$rs, \$rt
Slt	R	Slt \$rd, \$rs, \$rt
Nop	R	Nop \$rd, \$rs, \$rt
Lw	I	Lw \$rd, \$rs imm
Sw	I	Sw \$rd, \$rs imm
Slti	I	Slti \$rd, \$rs imm
Beq	I	Beq \$rd, \$rs imm