CS2201 Arquitectura de Computadoras

Proyecto Final

Fecha de entrega: Domingo 30 de Junio 2018 11:59pm

Desarrollar los problemas en Verilog
Enviar los códigos en una carpeta comprimida .tar.gz a rbustamante@utec.edu.pe (No se aceptarán carpetas comprimidas en .zip , .rar u otro formato) Titulo del email: LAB3
El nombre de la carpeta debe tener la primera letra de su nombre y su apellido. Ejm. rbustamante_proyectofinal.tar.gz
El proyecto puede realizarse en grupos de 2.

Para el proyecto Final se debe generar un informe en .pdf utilizando LaTex siguiendo el formato IEEE para conferencias: https://www.ieee.org/conferences/publishing/templates.html
El informe debe estar en INGLES y contener los siguientes puntos:

- Abstract
- Introduction
- Methodology
- Experimental Setup
- Evaluation
- Conclusions
- Comments (difultades encontradas, sugerencias, etc)

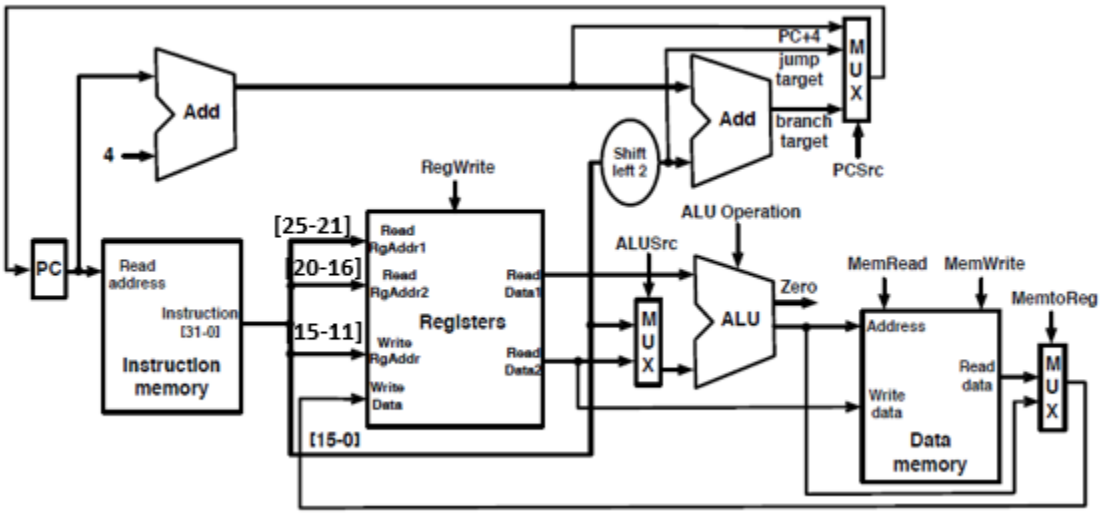
Desarrollar lo siguiente:

Tarea básica (18pts)

Escribir código Verilog para implementar un Pathline de 32 Bits RISC MIPS, que pueda soportar las siguientes instrucciones.

add	addi	lb	beq
sub	subi	lh	bneq
and	andi	lw	bgez
nor	ori	sb	j
or	slti	sh	jr
slt		sw	jal
		lui	

NAME	NUMBER	USE	PRESERVED ACROSS A CALL?
$zero	0	The Constant Value 0	N.A.
$at	1	Assembler Temporary	No
$v0-$v1	2-3	Values for Function Results and Expression Evaluation	No
$a0-$a3	4-7	Arguments	No
$t0-$t7	8-15	Temporaries	No
$s0-$s7	16-23	Saved Temporaries	Yes
$t8-$t9	24-25	Temporaries	No
$k0-$k1	26-27	Reserved for OS Kernel	No
$gp	28	Global Pointer	Yes
$sp	29	Stack Pointer	Yes
$fp	30	Frame Pointer	Yes
$ra	31	Return Address	Yes



El tamaño del Instruction Memory y del Data Memory es de 256 bytes.
Las memorias son byte-addressable (1byte)
El datapath es big-endian.
Para cargar data en la memoria de instrucciones usar $readmemh.
Todas las instrucciones deben funcionar.
Para probar las instrucciones deben cargarse 3 archivos a las memoria de programa por separado para verificar que las instrucciones han sido implementadas correctamente. A su vez cada programa debe tener un testbench que capture el estado de los registros que han sido utilizados por la instrucción.
En el primer programa debe ejecutarse todas las operaciones de las 2 primeras columnas. (Previamente poner valores al azar en $s0-$s7 $t0-$t9)
El segundo programa deben ejecutarse todas las operaciones de la 3ra columna. Previamente cargar numero al azar en la memoria de data.

En el tercer programa deben ejecutarse todas las instrucciones de la 4ta columna. Previamente realizar un programa que contenga instrucciones en el medio y etiquetas para verificar que realmente está cumpliendo beq, begz , bneq y j.

De C a ASM (2 pts)

Traduzca el código de C Assembler MIPS, cargarlo en la memoria de programa y

```
int fact(int n){
        if (n<1)
        return 1;
        else
        return n*factorial(n-1)
        }
variable = factorial(10);
```

variable se encuentra en $s0, por lo que $s0 al final del programa debe darnos el resultado del factorial de 10.

Puntos Extra + 4 pts en el examen final.

Modificar el datapath para que se convierta en un pipeline de 5 etapa; IF, ID, EXE, MEM, WB.
El pipeline debe soportar bubbles (nop)
Unidad de by-pass.
Añadir unidad de comparación rápida y de computación de dirección en la etapa de decodificación.