Universidad del Valle de Guatemala
Facultad de Ingeniería
Departamento de Ingeniería Electrónica
IE2011 - Electrónica Digital I
Kurt Kellner

Laboratorio #10

Este laboratorio será trabajado de forma individual y se entregará de forma digital de acuerdo a la fecha de entrega en Canvas. Deberá identificar su entrega con su nombre, carné y sección. Deberá adjuntar su código (los archivos `.v`), los diagramas de timing (tanto imágenes como los archivos `.vcd`), breves explicaciones en un archivo PDF y un link a su repositorio (donde debe haber cargado su código también). Para cada ejercicio deberá hacer un folder separado.

Ejercicio 01

Utilice los módulos que ha trabajado en laboratorios anteriores e implemente el siguiente circuito:

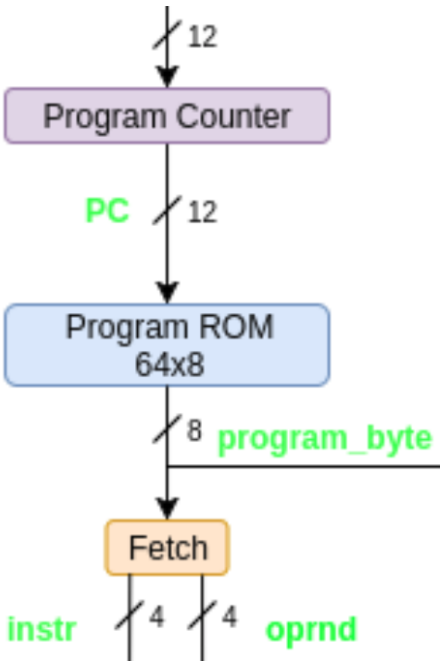

Figure 1: Ejercicio 01

Su módulo deberá tener como entradas *reset*, *clock*, los 12 bits del *Program Counter* para hacer *loads* y los bits necesarios para operar los módulos *Program Counter* y *Fetch* (*enabled* y *load*). Sus salidas deberán ser los 8 bits del *program_byte*, 4 bits de *intsr* y 4 bits de *oprnd*.

Implemente un testbench para probar la funcionalidad de todo el circuito. Puede utilizar los valores que usted desee dentro de la *Program ROM*. Recuerde entregar los diagramas de ***timing***.

Ejercicio 02

Utilice los módulos que ha trabajado en laboratorios anteriores e implemente el siguiente circuito:

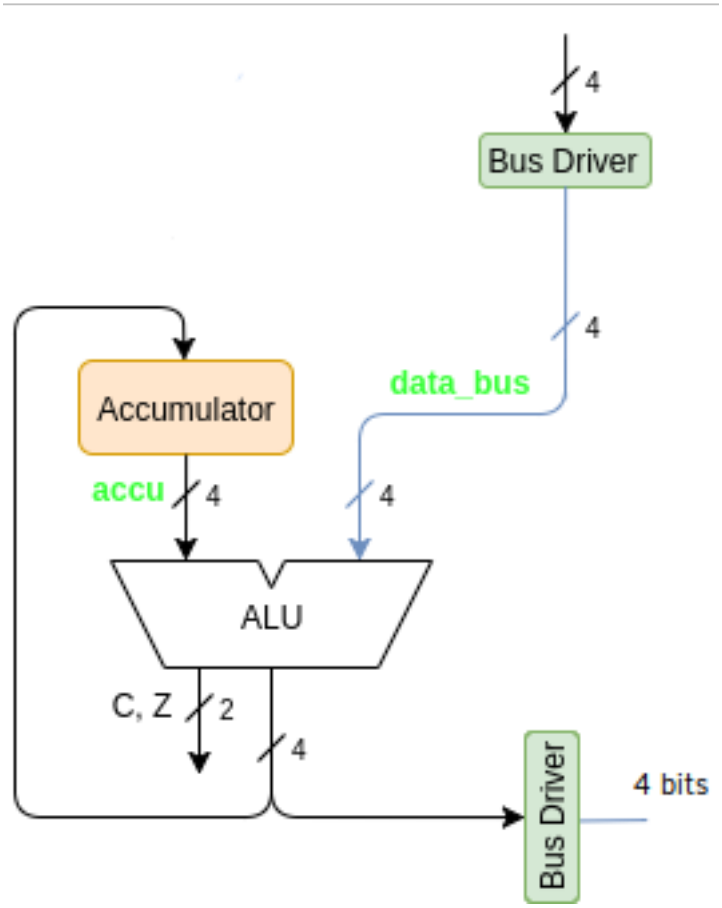

Figure 2: Ejercicio 02

Considere que la ALU no es la misma que en los laboratorios anteriores sino que debe ser la del proyecto. Como referencia la tabla de instrucciones del procesador completo es esta:

Instrucción	i3	i2	i1	i0	C	Z	phase	incPC	loadPC	loadA	loadFlags	S2	S1	S0	csRAM	weRAM	oeALU	oeIN	oeOprnd	loadOut
any	x	x	x	x	x	x	0	1	0	0	0	0	0	0	0	0	1	0	0	0
JC	0	0	0	0	1	x	1	0	1	0	0	0	0	0	0	0	1	0	0	0
JC	0	0	0	0	0	x	1	1	0	0	0	0	0	0	0	0	1	0	0	0
JNC	0	0	0	1	1	x	1	1	0	0	0	0	0	0	0	0	1	0	0	0
JNC	0	0	0	1	0	x	1	0	1	0	0	0	0	0	0	0	1	0	0	0
CMPI	0	0	1	0	x	x	1	0	0	0	1	0	0	1	0	0	0	0	1	0
CMPM	0	0	1	1	x	x	1	1	0	0	1	0	0	1	1	0	0	0	0	0
LIT	0	1	0	0	x	x	1	0	0	1	1	0	1	0	0	0	0	0	1	0
IN	0	1	0	1	x	x	1	0	0	1	1	0	1	0	0	0	0	1	0	0
LD	0	1	1	0	x	x	1	1	0	1	1	0	1	0	1	0	0	0	0	0
ST	0	1	1	1	x	x	1	1	0	0	0	0	0	0	1	1	1	0	0	0
JZ	1	0	0	0	x	1	1	0	1	0	0	0	0	0	0	0	1	0	0	0
JZ	1	0	0	0	x	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0
JNZ	1	0	0	1	x	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0
JNZ	1	0	0	1	x	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0
ADDI	1	0	1	0	x	x	1	0	0	1	1	0	1	1	0	0	0	0	1	0
ADDM	1	0	1	1	x	x	1	1	0	1	1	0	1	1	1	0	0	0	0	0
JMP	1	1	0	0	x	x	1	0	1	0	0	0	0	0	0	0	1	0	0	0
OUT	1	1	0	1	x	x	1	0	0	0	0	0	0	0	0	0	1	0	0	1
NORI	1	1	1	0	x	x	1	0	0	1	1	1	0	0	0	0	0	0	1	0
NORM	1	1	1	1	x	x	1	1	0	1	1	1	0	0	1	0	0	0	0	0

Figure 3: Tabla de Instrucciones

Implemente un testbench para probar la funcionalidad del circuito. En este caso necesitará las señales de *clk* y *reset* para el *Accumulator*, señales de control para los buffers tri-estado y las señales de control para la ALU. Las salidas **C** y **Z** de la ALU quieren decir ***Carry*** y ***Zero***. La señal de ***Zero*** se enciende cuando todos los bits de salida de la ALU son 0s. La señal de ***Carry*** se enciende cuando la operación realizada tiene un *overflow*. Recuerde incluir sus diagramas de timing.