**Universidad Nacional Autónoma de México**

**Facultad de Ingeniería**

**Diseño de Sistemas Digitales**

Aguilar Enriquez Paul Sebastian – 415028130

Cabrera López Oscar Emilio - 312333261

2016/11/23

**Examen Segundo Parcial**

**Descripción del problema**

Diseñar y montar un circuito que simule un cambio automático de velocidades.

El circuito deberá contar con un “acelerador” al cual deberán responder dos motores que simulen el movimiento automático de una palanca de velocidades.

**Método de solución**

Para el desarrollo de esta práctica se decidió utilizar componentes en físico así como el FPGA Cyclone II.

El problema se abordó dividiéndolo en “módulos”, tanto en físico como en el FPGA.

Los módulos que se diseñaron e implementaron en físico fueron:

* Control de velocidad de un motor de CD con un TIP31.
* Detector de revoluciones con un sensor de herradura H21A1 y una compuerta AND.
* ADC (con el CI ADC0844) para medir el voltaje regulado en el control de velocidad que posteriormente se enviara como dato al FPGA.
* Dos motores a pasos de tipo bipolar.

Los módulos que se diseñaron e implementaron en el FPGA fueron:

* Detector de frecuencias para determinar la velocidad actual a partir de las RPM del motor.
* Máquina de estados para simular los cambios automáticos de velocidades.
* Control de movimiento de los motores a pasos.

*Implementación*

Se implementó un control de velocidad de un motor de CD utilizando un TIP31, este módulo corresponde al “acelerador” requerido en el planteamiento del problema. Al motor que se estaba regulando se le coloco una pestaña de cartón la cual pasa por el sensor de herradura H21A1.

El sensor de herradura detecta la pestaña que esta adjunta al motor, la salida del receptor del sensor se conecta a una compuerta AND que tiene como entrada un “uno”, así cuando la señal del receptor esta en alto (que detecto la pestaña) se produce como salida de la compuerta AND un “uno” lógico, esto nos permite ir generando un tren de pulsos que corresponden a las revoluciones del motor.

**El tren de pulsos generado en el módulo anterior entra al FPGA el cual…**

Se utilizó un ADC0844, el cual se usa para convertir el voltaje que es regulado en el módulo del control de velocidad del motor de CD a un valor digital (en binario) el cual se manda como dato al FPGA para determinar si el acelerador está acelerando o desacelerando.

**Cuando esta señal es recibida en el FPGA…**

A su vez como “control automático de velocidades” se implementó una máquina de estados en lenguaje VHDL mediante cartas ASM, resultando ser una maquina Mealy. Esta máquina de estados responde a las revoluciones actuales del motor para determinar la velocidad, si sale del rango establecido para una velocidad se cambia a la velocidad correspondiente, esto considerando un control de ventana para cada velocidad. Si hay un cambio de estado, el estado actual (el mas nuevo) se manda al control de movimiento de los motores a pasos como la velocidad a la quebe cambiar la palanca.

**El control de movimiento para dos motores a pasos los cuales corresponden al movimiento de la palanca de velocidades (eje X y eje Y) también se implementó en VHDL …**

**Modelado de las variables internas**

Para modelar las variables decidimos hacer un diagrama de cartas ASM quedando de la siguiente manera.



**Código**

-- Copyright (c) 2016

-- Emilio Cabrera <emilio1625@gmail.com>

-- Sebastian Aguilar <penserbjorne>

--

-- GNU GENERAL PUBLIC LICENSE

-- Version 3, 29 June 2007

--

-- This program is free software: you can redistribute it and/or modify

-- it under the terms of the GNU General Public License as published by

-- the Free Software Foundation, either version 3 of the License, or

-- (at your option) any later version.

--

-- This program is distributed in the hope that it will be useful,

-- but WITHOUT ANY WARRANTY; without even the implied warranty of

-- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

-- GNU General Public License for more details.

--

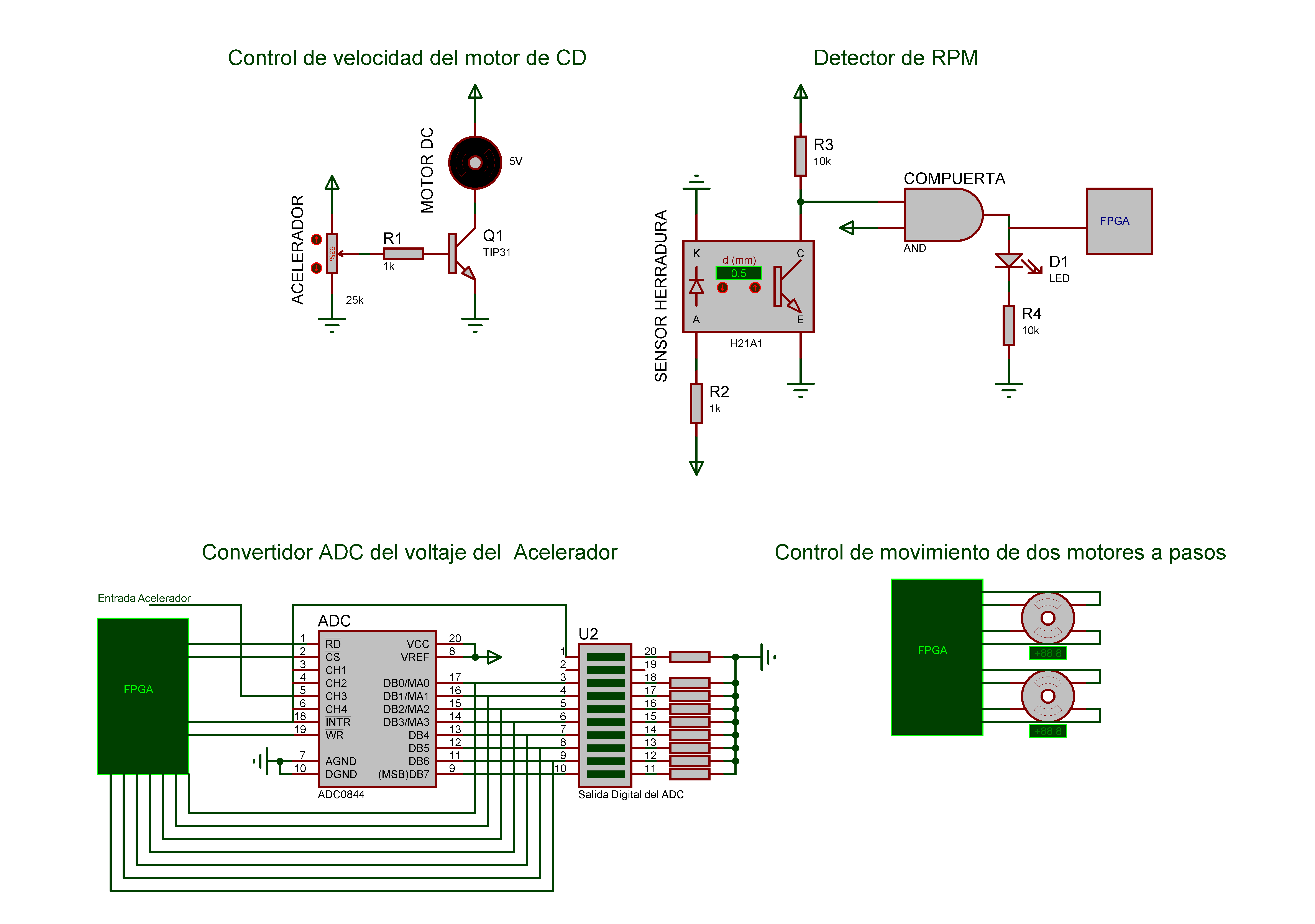
-- You should have received a copy of the GNU General Public License

-- along with this program. If not, see <http://www.gnu.org/licenses/>.

*Archivo del módulo del temporizador “tock.vhd”*

library ieee;

**Construcción virtual**

****

**Conclusiones**

* Aguilar Enriquez Paul Sebastian: El proyecto se logró completar de manera satisfactoria, sin embargo hasta ahora ha sido el más complicado de los tres, y no en la cuestión de la materia (diseño de sistemas digitales),si no que en general tardamos bastante en desarrollar todo lo que era ajeno a lo digital o a lo visto en clase, nos volvimos a encontrar con pequeños detalles que tardamos bastante tiempo en solucionar como el control de la velocidad del motor DC o el lograr conectar correctamente el ADC, sobre todo la parte de acoplamiento con el FPGA, en general eran detalles pequeños pero que debido a la poca practica que tenemos en el área nos tomaron más tiempo del deseado. En general el desarrollo del proyecto fue muy productivo y nos permitió ahondar en otros temas de interés como el manejo de motores y la implementación de una máquina de estado más compleja.
* Cabrera López Oscar Emilio: Que chingue a su madre el América.

**Repositorio en GitHub** <https://github.com/penserbjorne/clase-diseniosistemasdigitales-2017-1>