



UTN – Departamento de Electrónica
Cátedra : Robótica

Trabajo Práctico N° 2

Análisis dinámico de un robot e
implementación en FPGA

INTEGRANTES:

BALEA, MIGUEL ALBERTO
MURUAGA, LEONARDO SEBASTIÁN

PROFESOR:

HERNÁN GIANNETTA

FECHA DE ENTREGA: 10 / JULIO / 2009

CURSO:

R6055

AÑO:

2009

Índice temático de contenidos

Índice	página 2
Introducción	página 3
Robot: FIVE BARS LINKAGE	página 4
Análisis dinámico	página 4 a 6
Expresiones finales del torque	página 7
Gráfico de posición en f. del tiempo	página 6
Gráfico de posición, velocidad y aceleración	página 7
Gráfico de torque de motor q1	página 7
<u>Conclusiones del modelo dinámico</u>	página 8
Motor: BRUSHLESS	página 9
Descripción de la lógica a utilizar	página 9 a 10
Código de la “llave”	página 10 a 11
Código del “test bench”	página 11 a 14
<u>Conclusiones del diseño VHDL</u>	página 14
<u>Simulación de las señales del motor y puente</u>	página 15

Introducción:

La dinámica es la parte de la física que describe la evolución en el tiempo de un sistema físico en relación con las causas que provocan los cambios de estado físico y/o estado de movimiento.

El objetivo de la dinámica es describir los factores capaces de producir alteraciones de un sistema físico, cuantificarlos y plantear ecuaciones de movimientos o ecuaciones de evolución para dicho sistema.

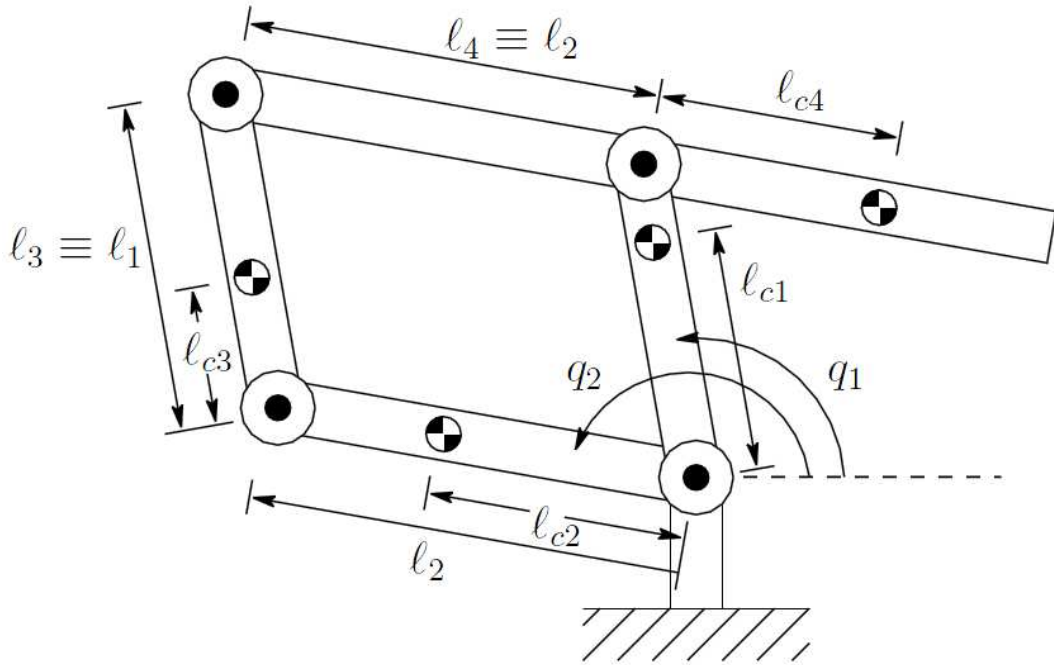
El estudio de la dinámica es prominente en los sistemas mecánicos.

Por lo tanto, el modelo dinámico de un robot tiene por objetivo conocer la relación entre el movimiento del robot y las fuerzas implicadas en el mismo.

Esta relación se obtiene mediante el denominado modelo dinámico, que relaciona matemáticamente:

- La localización del robot definida por sus variables articulares o por las coordenadas de localización de su extremo, y sus derivadas: velocidad y aceleración.
- Las fuerzas y pares aplicados en las articulaciones.
- Los parámetros dimensionales del robot, como longitud, masas e inercias de sus elementos.

Robot: FIVE BAR LINKAGE



En el presente trabajo práctico se obtendrá el modelo dinámico del motor articulado de 5 barras y se determinará a partir este modelo los torques involucrados en los motores para una cierta trayectoria dada.

Como objetivo de lo anterior se esperará determinar el máximo torque que deberán realizar los motores y por consiguiente determinar el torque nominal de ellos.

Análisis dinámico:

Para nuestra simulación se consideró que

$l_1 = 1m$	$l_{c1} = 0,5m$	$m_1 = 0,6kg$
$l_2 = 0,05m$	$l_{c2} = 0,025m$	$m_2 = 0,3kg$
$l_3 = 1m$	$l_{c3} = 0,4m$	$m_3 = 0,25kg$
$l_4 = 0,1m$	$l_{c4} = 0,05m$	$m_4 = 0,2kg$

Las siguientes expresiones nos permiten calcular las componentes de la matriz de inercia del robot. Se obtuvieron aplicando el método energético de Lagrange – Euler.

$$\begin{aligned}
 d_{11}(q) &= m_1 \ell_{c1}^2 + m_3 \ell_{c3}^2 + m_4 \ell_1^2 + I_1 + I_3 \\
 d_{12}(q) &= d_{21}(q) = (m_3 \ell_2 \ell_{c3} - m_4 \ell_1 \ell_{c4}) \cos(q_2 - q_1) \\
 d_{22}(q) &= m_2 \ell_{c2}^2 + m_3 \ell_2^2 + m_4 \ell_{c4}^2 + I_2 + I_4
 \end{aligned}$$

En donde puede observarse claramente como los términos de la matriz dependen de los momentos de inercia de cada barra respecto de su centro de masa y del desplazamiento de ellos respecto al eje de rotación (lo que se condice con el Teorema de Steiner que permite calcular el momento de inercia con respecto a un eje que no pasa por el centro de masa).

Si se cumple con la relación

$$m_3 \ell_2 \ell_{c3} = m_4 \ell_1 \ell_{c4}$$

Se anulan los términos d_{12} y d_{21} ya que los componentes mecánicos del robot (barras) no presentan rotaciones al actuar los motores q_1 y q_2 .

Finalmente se obtiene que los torques de los motores q_1 y q_2 están dados por

$$d_{11} \ddot{q}_1 + \phi_1(q_1) = \tau_1, \quad d_{22} \ddot{q}_2 + \phi_2(q_2) = \tau_2$$

Siendo d_{11} y d_{22} las componentes de la diagonal principal de la matriz de inercia y Φ_1 y Φ_2

$$\phi_1 = g \cos q_1 (m_1 \ell_{c1} + m_3 \ell_{c3} + m_4 \ell_1)$$

$$\phi_2 = g \cos q_2 (m_2 \ell_{c2} + m_3 \ell_2 - m_4 \ell_{c4})$$

Para nuestro robot se tendrá entonces que

$$d_{11} = 0,6kg \times (0,5m)^2 + 0,25kg \times (0,4m)^2 + 0,2kg \times (1m)^2 + I_1 + I_3$$

$$d_{11} = 0,39kg \times m^2$$

$$d_{22} = 0,2kg \times (0,025m)^2 + 0,25kg \times (0,05m)^2 + 0,2kg \times (0,05m)^2 + I_2 + I_4$$

$$d_{22} = 0,00125kg \times m^2$$

En donde se consideraron despreciables I_1 , I_2 , I_3 e I_4 ya que no se conoce la geometría ni las dimensiones de las barras que componen el robot y sólo afectan en el sentido de aumentar los términos de la matriz.

Y además

$$\Phi_1 = 9,8 \frac{m}{s^2} \cos(q_1) (0,6kg \times 0,5m + 0,25kg \times 0,4m + 0,2kg \times 1m)$$

$$\Phi_1 = 5,88kg \frac{m^2}{s^2} \cos(q_1)$$

$$\Phi_2 = 9,8 \frac{m}{s^2} \cos(q_2) (0,3kg \times 0,025m + 0,25kg \times 0,05m - 0,2kg \times 0,05m)$$

$$\Phi_2 = 0,098kg \frac{m^2}{s^2} \cos(q_2)$$

Por lo tanto los torques para los motores q1 y q2 quedarán determinados por

$$\tau_1 [N \times m] = 0,39 \times \ddot{q}_1 + 5,88 \times \cos(q_1) \times q_1$$

$$\tau_2 [N \times m] = 0,00125 \times \ddot{q}_2 + 0,098 \times \cos(q_2) \times q_2$$

Supongamos entonces una variación temporal senoidal en donde la posición de q1 varíe de 60 a 15 grados en 2 segundos.

La siguiente ecuación representa la posición de la rotación q1 en función del tiempo.

$$q_1 = \frac{5}{24} \pi + \frac{\pi}{8} \sin\left(\frac{\pi}{2s} \times t + \frac{\pi}{2}\right)$$

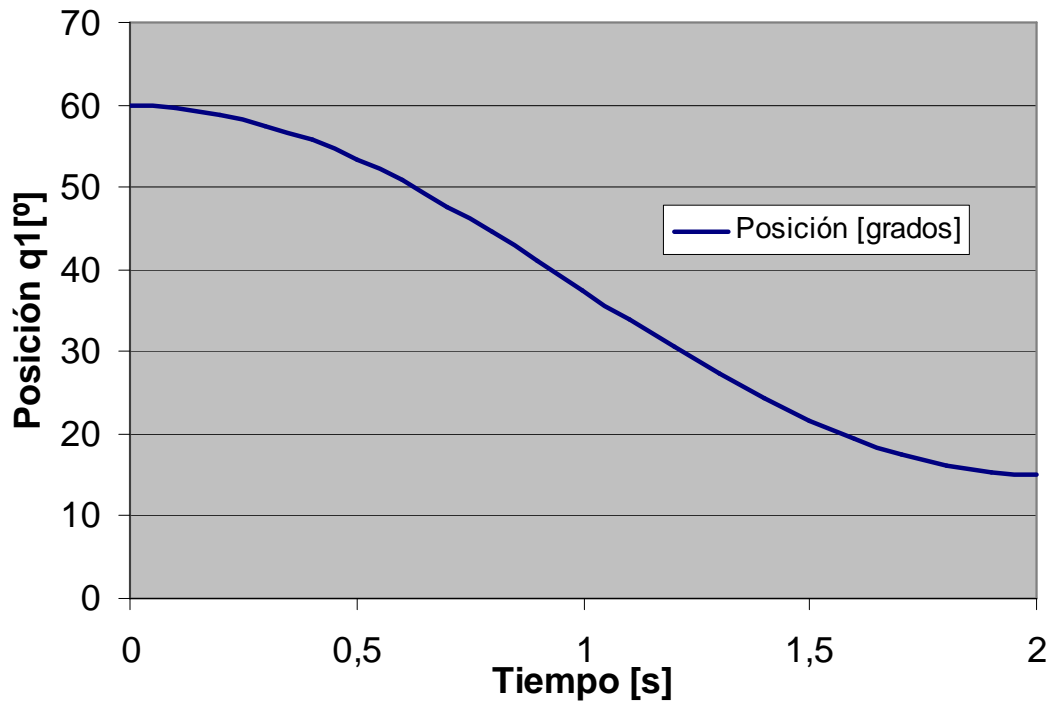


Gráfico de la posición deseada en función del tiempo

Se representan a continuación la posición en radianes y los gráficos de velocidad y aceleración.

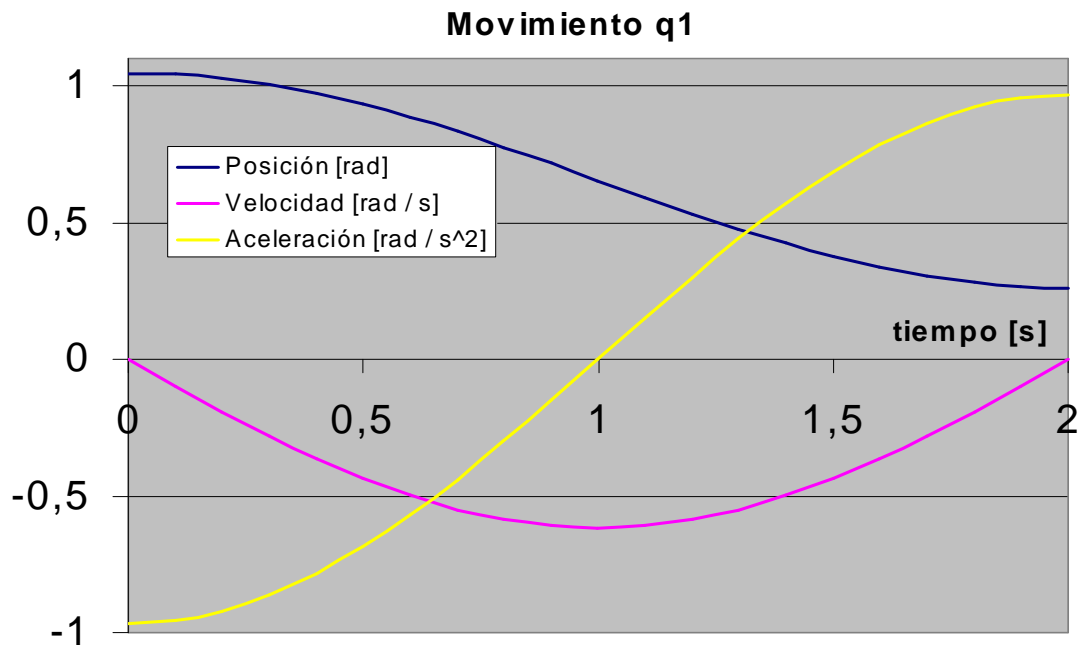
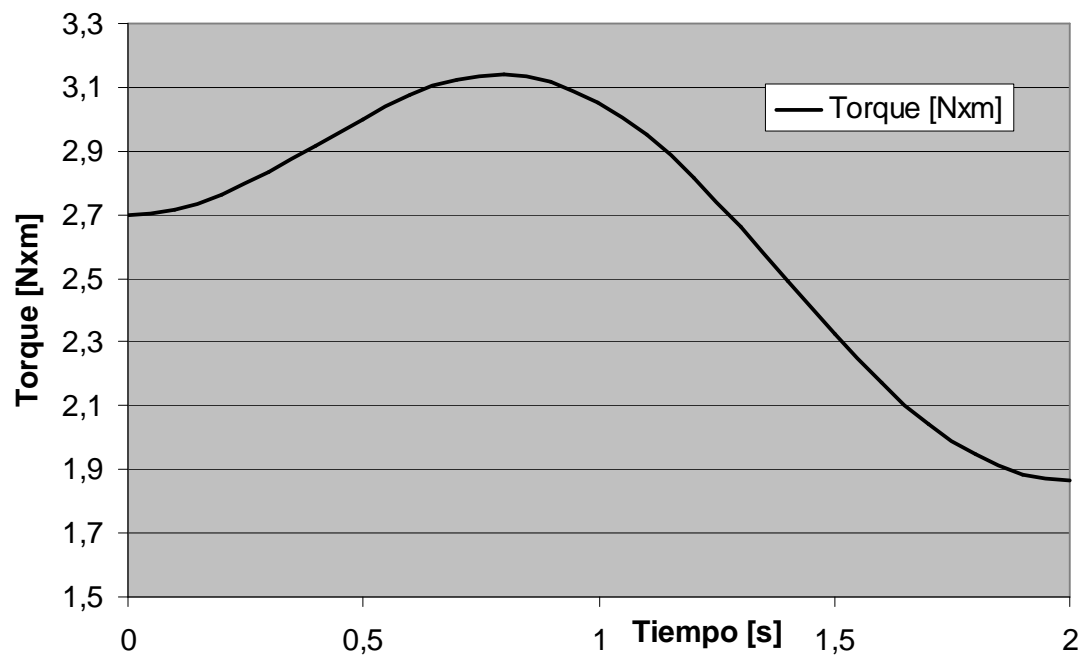


Gráfico de la posición, velocidad y aceleración angular

Determinemos entonces el torque requerido en el motor q1 para realizar el movimiento en la condiciones establecidas anteriormente.



Torque del motor q1 en función del tiempo para las condiciones del movimiento

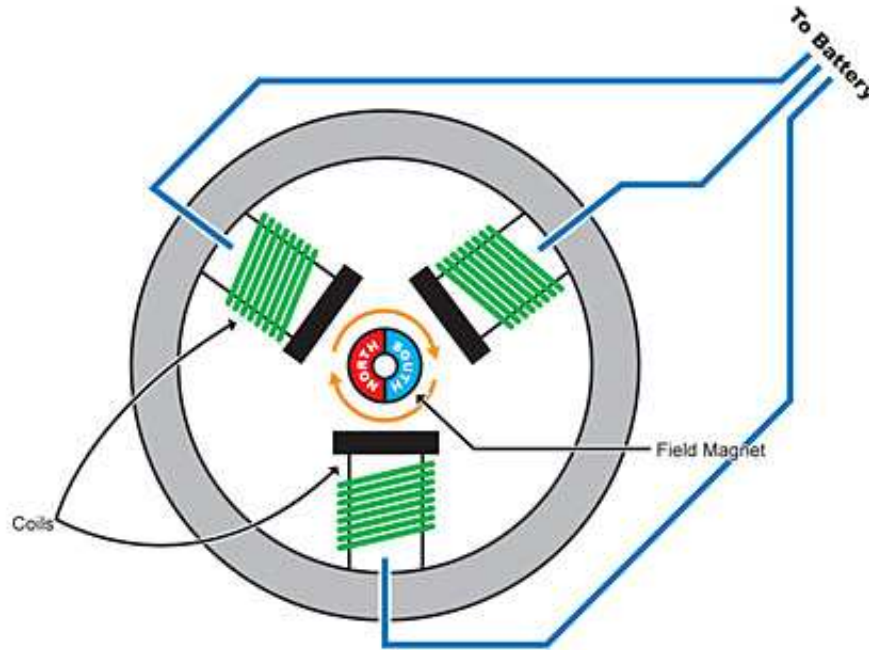
Conclusiones del estudio dinámico:

Debido a la geometría del robot y a las masas de las barras se observa que el motor q1 debe tener mayor capacidad de torque que el motor q2 ya que los momentos de inercia que debe vencer son superiores por las distancias que están involucran entre los centros de masa y el eje de rotación.

Además en el último gráfico se observa que el torque Pico que se requiere para seguir el movimiento establecido es de 3,1 Nm; si se eligiera un motor de corriente continua tipo Brushless (sin escobillas) se necesitaría un torque nominal de 1,2 Nm.

A pesar de que en el movimiento elegido se escogió la forma senoidal, esperando disminuir las variaciones de aceleración, se observa que la aceleración varía bruscamente al iniciar y al finalizar el movimiento, y el torque, a pesar de tener diferente forma, también varía bruscamente al inicio y fin del movimiento. En el caso práctico de tener que utilizar este robot sería conveniente que las aceleraciones partieran desde cero, por lo que sería conveniente elegir la forma de la velocidad senoidal, en vez de la forma de la posición.

Motor: BRUSHLESS

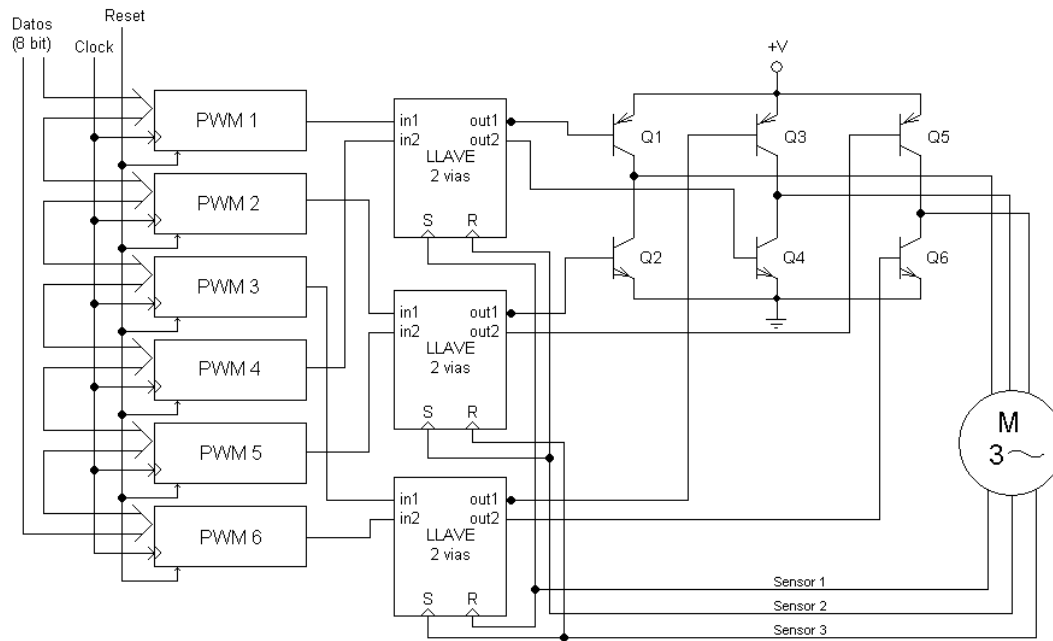


En esta etapa del trabajo práctico se procederá a diseñar el sistema de modulación PWM que se utilizará para controlar al motor brushless del robot anterior. Se desarrollará este sistema haciendo uso de lenguaje descriptor de hardware (VHDL) y de FPGA.

El objetivo es obtener las seis señales necesarias para gobernar un puente H trifásico en función de la velocidad requerida, y haciendo uso de los sensores de posición de efecto Hall presentes en el motor, los que dan idea de la posición del rotor y que facilitan el conocer cuando deben ser activados los diferentes arrollamientos a favor de obtener el máximo par.

En la siguiente página se observa la lógica que se pretende diseñar. Las señales de datos, clock y reset provienen del dispositivo encargado de controlar el motor (generalmente microcontrolador). Por una cuestión de simplicidad en el gráfico no se representaron independientemente los seis datos diferentes que controlarán a cada modulador PWM.

Luego de ser generadas las seis señales PWM son tomadas de a pares e introducidas en tres llaves digitales de 2 vías, las cuales irán permitiendo el paso de las señales hacia el puente en función de la posición del rotor, lo que se obtiene con los tres sensores.



Utilizando como base una nota de aplicación de la firma ATMEL, se generaron seis instancias del modulador PWM.

Además se desarrolló la entidad LLAVE; una arquitectura que presenta dos entradas, dos salidas y dos señales de set y reset por flanco ascendente. Cuando se presenta un flanco positivo en la entrada de set, el dispositivo se habilita, y las salidas copian a las entradas. Al presentarse un flanco positivo en el reset, el dispositivo se apaga y ambas salidas son conmutadas a cero.

Se aprovecharon las tres señales de los sensores para activar / desactivar cada llave y así encender o no cada camino del puente H trifásico y por consiguiente controlar el motor.

Código VHDL de la llave

Para el diseño de la llave se generaron dos procesos, el primero es el encargado de determinar los flancos y generar la señal de habilitación y el segundo permite o no el paso de las señales a las salidas.

```
library IEEE;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all ;

ENTITY llave IS
  PORT
  ( in1, in2, set, rst  :in STD_LOGIC;
    out1, out2          :out STD_LOGIC
  );
END llave;

ARCHITECTURE l2vias OF llave IS

  SIGNAL set_on, out1_int, out2_int  : STD_LOGIC;

BEGIN
  --En función del estado del latch deja pasar las señales
  --o anula las salidas
```

```

PROCESS (in1, in2, set_on, out1_int, out2_int)
BEGIN
    IF (set_on = '1') THEN
        out1_int <= in1;
        out2_int <= in2;
    ELSE
        out1_int <= '0';
        out2_int <= '0';
    END IF;
    out1 <= out1_int;
    out2 <= out2_int;
END PROCESS;

--Proceso de detección de flancos y activación / desactivación del
latch "set_on"
PROCESS (set, rst, set_on)
BEGIN
    IF (rising_edge(set)) THEN
        set_on <= '1';
    ELSIF (rising_edge(rst)) THEN
        set_on <= '0';
    END IF;
END PROCESS;

END l2vias;

```

Código VHDL del test bench:

Este archivo tiene como función generar las señales de reloj, de los sensores y de datos para poder simular la lógica de la figura de la página anterior. Está basado en el “test_bench” proporcionado por ATMEL.

Lamentablemente debido a problemas con el simulador se utilizó un solo generador PWM.

```

LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all ;
USE work.user_pkg.all;

ENTITY test_bench IS

END test_bench;

ARCHITECTURE arch_test_bench OF test_bench IS

COMPONENT pwm_fpga
PORT (
    clock: in std_logic;
    reset: in std_logic;
    data_value: in std_logic_vector(7 downto 0);
    pwm: out std_logic
);
END COMPONENT;

COMPONENT llave
PORT (
    in1 :in std_logic;
    in2 :in std_logic;

```

```

        set :in std_logic;
        rst :in std_logic;
        out1 :out std_logic;
        out2 :out std_logic
    );
END COMPONENT;

-- Declaración de señales
SIGNAL sig_sen1 : std_logic;
SIGNAL sig_sen2 : std_logic;
SIGNAL sig_sen3 : std_logic;
SIGNAL sig_pwm1 : std_logic;
SIGNAL sig_pwm2 : std_logic;
SIGNAL sig_pwm3 : std_logic;
SIGNAL sig_pwm4 : std_logic;
SIGNAL sig_pwm5 : std_logic;
SIGNAL sig_pwm6 : std_logic;
SIGNAL sig_out11 : std_logic;
SIGNAL sig_out14 : std_logic;
SIGNAL sig_out22 : std_logic;
SIGNAL sig_out25 : std_logic;
SIGNAL sig_out33 : std_logic;
SIGNAL sig_out36 : std_logic;
SIGNAL sig_clock : std_logic;
SIGNAL sig_reset : std_logic;
SIGNAL sig_data_value : std_logic_vector(7 downto 0);
SIGNAL value : std_logic_vector(7 downto 0);
SIGNAL sig_pwm : std_logic;
shared variable ENDSIM: boolean:=FALSE;
    constant clk_period:TIME:=100 ns;
    constant t:TIME:=250us;

BEGIN
--Aquí se genera el reloj del sistema
clk_gen: process
    BEGIN
        If ENDSIM = FALSE THEN
            sig_clock <= '1';
            wait for clk_period/2;
            sig_clock <= '0';
            wait for clk_period/2;
        else
            wait;
        end if;
    end process;

--Aquí se generan las señales de los 3 sensores de posición Hall
sen_gen: process
    BEGIN
        sig_sen1 <= '1';
        wait for clk_period * 20000 /6;
        sig_sen1 <= '0';
        wait for clk_period * 20000 /6;
        sig_sen2 <= '1';
        wait for clk_period * 20000 /6;
        sig_sen2 <= '0';
        wait for clk_period * 20000 /6;
        sig_sen3 <= '1';
        wait for clk_period * 20000 /6;
        sig_sen3 <= '0';
        wait for clk_period * 20000 /6;
    
```

```
end process;

-- Generación de los componentes "pwm_fpga"
inst_pwm_fpga1 : pwm_fpga
PORT MAP(
    clock => sig_clock,
    reset => sig_reset,
    data_value => sig_data_value,
    pwm    => sig_pwm1
);

-- Generación de los componentes "llave"
inst_llave1 : llave
PORT MAP(
    set => sig_sen1,
    rst => sig_sen2,
    in1 => sig_pwm1,
    in2 => sig_pwm2,
    out1 => sig_out11,
    out2 => sig_out14
);

inst_llave2 : llave
PORT MAP(
    set => sig_sen2,
    rst => sig_sen3,
    in1 => sig_pwm1,
    in2 => sig_pwm2,
    out1 => sig_out22,
    out2 => sig_out25
);

inst_llave3 : llave
PORT MAP(
    set => sig_sen3,
    rst => sig_sen1,
    in1 => sig_pwm1,
    in2 => sig_pwm2,
    out1 => sig_out33,
    out2 => sig_out36
);

--Aquí se resetean los PWM y se genera la rampa de velocidad
stimulus_process: PROCESS
BEGIN
    sig_pwm2 <= '1';
    sig_reset <= '1';
    wait for 50us;
    sig_data_value <= "11111111";
    sig_reset <= '0';
    wait for 1ms;
    sig_data_value <= "11111111";
    wait for t;
    sig_data_value <= "11101011";
    wait for t;
    sig_data_value <= "11010111";
    wait for t;
    sig_data_value <= "11000011";
    wait for t;
    sig_data_value <= "10101111";
```

```
wait for t;
sig_data_value <= "10011011";
wait for t;
sig_data_value <= "10000111";
wait for t;
sig_data_value <= "01110011";
wait for t;
sig_data_value <= "01011111";
wait for t;
sig_data_value <= "01001011";
wait for t;
sig_data_value <= "00110111";
wait for t;
sig_data_value <= "00100011";
wait for t;
sig_data_value <= "00001111";
wait for t;
wait;
END PROCESS stimulus_process;
END arch_test_bench;
```

De la simulación anterior se obtienen las señales que se muestran en la próxima página.

Las primeras 3 señales son las respectivas a los sensores de posición Hall.

Las próximas 6 tomadas de a pares se corresponden con las salidas de cada una de las tres llaves que son las que activarán los tres caminos del puente H.

La última señal es la salida del modulador de PWM. Como puede apreciarse esta señal está generada basándose en una rampa de velocidad, lo que se aprecia fácilmente en la variación temporal del ancho de pulso.

Conclusión:

En la segunda parte del trabajo práctico, nos enfrentamos con los problemas que se presentan al tratar de implementar el control de motor brushless con programación de hardware. A pesar de tener las grandes ventajas de la altísima velocidad de conmutación, lo que permite realizar operaciones en tiempos extremadamente pequeños en comparación con un microcontrolador, tiene la desventaja de que el diseño debe partir de la basa misma de la lógica y debe tener en cuenta no solo lo que se trata de implementar, sino también todas las condiciones iniciales de las salidas para los casos en que no se hayan definido las entradas.

Por otra parte, y a pesar de que no pudieron generarse las seis instancias del modulador PWM, creemos que el resultado fue satisfactorio porque se generaron las señales de control del puente y la rampa de velocidad que se esperaba obtener.

Lamentablemente, la simulación no representa la realidad del motor ya que las señales de los sensores que aparecen igualmente espaciadas y en la realidad esta separación disminuiría al incrementarse la velocidad de rotación.

