PDS-FPGAs

Práctica P1
Grupo 09

Moreno Suay, Sergio
Ruiz Escorihuela, Víctor Javier

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Índice

1. P1.1: Implementación del módulo DDS

1.1. Descripción del módulo

ENUNCIADO 1

Breve explicación del bloque implementado indicando su funcionalidad y estructura interna. Se puede añadir (de apoyo a la explicación) una figura con el diagrama de bloques interno del modulo.

En esta práctica se ha implementado un Sintetizador de frecuencias (DDS), que tiene como objetivo generar señales con una forma de onda sinusoidal, rectagular y rampa, en este caso, y una frecuencia y resolución editables a partir de una frecuencia de reloj.

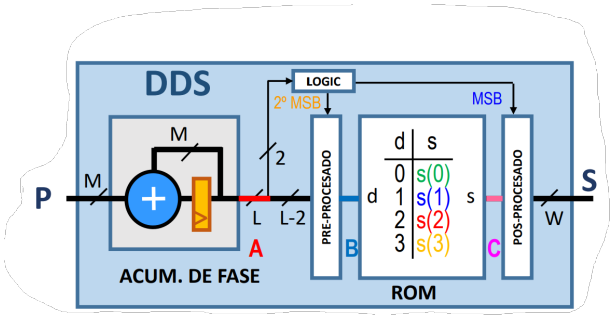


Figura 1: Esquema del bloque DDS.

Como se observa en la figura 1 este bloque consta de un acumulador, un bloque de pre-procesado, una ROM y un bloque de post-procesado, principalmente.

1.2. Interfaz

ENUNCIADO 2

Incluir las tablas en las que se definan el interfaz, sus formatos y sus parámetros. En esta práctica las tablas están por completar para que sirva de ejemplo de cómo hay que hacerlo.

Tabla 1: Parámetros del módulo DDS_test

NOMBRE	DESCRIPCIÓN
M	Tamaño del acumulador
L	Numero de bits del truncado de fase
W	Resolución de la ROM (ROM Wordlength)

Tabla 2: Interfaz del módulo DDS_test

NOMBRE	TIPO	FORMATO	DESCRIPCIÓN
id_p_ac	in	U[M,M]	Paso del acumulador
ic_rst_ac	in	bit	Reset síncrono del acumulador, activo a nivel alto
ic_en_ac	in	bit	Enable síncrono del acumulador, activo a nivel alto
ic_data_val	in	bit	Validación de muestras de entrada
clk	in	bit	Entrada de reloj
od_sin_wave	out	S[W,W-1]	Señal senoidal de frecuencia $f_o = P \cdot f_{clk}/2^M$
od_sqr_wave	out	S[W,W-1]	Señal cuadrada $f_o = P \cdot f_{clk}/2^M$
od_ramp_wave	out	S[W,W-1]	Señal rampa $f_o = P \cdot f_{clk}/2^M$
oc_data_val	out	bit	Señal de muestras validas de salida

1.3. Resultados de implementación

1.3.1. Recursos hardware

ENUNCIADO 3

Incluir:

- *Una tabla detallando los recursos del dispositivo FPGA que se requieren para implementar el módulo. En esta práctica solo hay que indicar el número de LEs configurados en modo normal, LEs en modo aritmético, los FFs y memorias M9K.*
- *Una captura del resumen de recursos obtenidos con la herramienta Quartus tras la etapa de emplazamiento y rutado (Fitter), que se muestra en el report "Resource Usage Summary". En esta primera memoria se ha incluido una captura de este report como ejemplo, que deberá sustituirla por la que se obtenga en su proyecto.*

El objetivo de esta sección es que el alumno sea capaz de identificar si los recursos obtenidos en la implementación son coherentes con los esperables para el circuito realizado.

Tabla 3: Recursos hardware

RECURSO	NÚMERO OBTENIDO
FFs	45
LEs-normal	37
LEs-aritméticos	38
M9Ks	16

A continuación se muestra la captura realizada de los recursos obtenidos tras la etapa de emplazamiento y rutado (Fitter) en el " Resource Usage Summary":

	Resource	Usage
1	Total logic elements	76 / 114,480 (< 1 %)
1	-- Combinational with no register	31
2	-- Register only	1
3	-- Combinational with a register	44
2		
3	Logic element usage by number of LUT inputs	
1	-- 4 input functions	0
2	-- 3 input functions	38
3	-- <=2 input functions	37
4	-- Register only	1
4		
5	Logic elements by mode	
1	-- normal mode	37
2	-- arithmetic mode	38
6		
7	Total registers*	45 / 117,053 (< 1 %)
1	-- Dedicated logic registers	45 / 114,480 (< 1 %)
2	-- I/O registers	0 / 2,573 (0 %)
8		
9	Total LABs: partially or completely used	8 / 7,155 (< 1 %)
10	Virtual pins	0
11	I/O pins	77 / 529 (15 %)
1	-- Clock pins	1 / 7 (14 %)
2	-- Dedicated input pins	0 / 9 (0 %)
12		
13	M9Ks	16 / 432 (4 %)
14	Total block memory bits	131,072 / 3,981,312 (3 %)
15	Total block memory implementation bits	147,456 / 3,981,312 (4 %)
16	Embedded Multiplier 9-bit elements	0 / 532 (0 %)
17	PLLs	0 / 4 (0 %)
18	Global signals	1
1	-- Global clocks	1 / 20 (5 %)
19	JTAGs	0 / 1 (0 %)
20	CRC blocks	0 / 1 (0 %)
21	ASMI blocks	0 / 1 (0 %)
22	Oscillator blocks	0 / 1 (0 %)
23	Impedance control blocks	0 / 4 (0 %)
24	Average interconnect usage (total/H/V)	0.1% / 0.1% / 0.1%
25	Peak interconnect usage (total/H/V)	2.2% / 2.1% / 2.3%
26	Maximum fan-out	61
27	Highest non-global fan-out	45
28	Total fan-out	703
29	Average fan-out	2.33

Figura 2: Report de “Resource Usage Summary”.

1.3.2. Frecuencia de operación

ENUNCIADO 4

La frecuencia máxima de operación del módulo implementado en el dispositivo y una indicación de dónde se encuentra el camino crítico del circuito. En la tabla aparece un ejemplo de cómo debe indicar el camino crítico.

Tabla 4: Frecuencia máxima de reloj

f_{clk} (MHz)	CAMINO CRÍTICO
196.23	Desde el registro de la etapa de pre-procesado al registro de la ROM

1.4. Verificación

ENUNCIADO 5

Resultados de la verificación al excitarlo con diferentes entradas para demostrar su completo funcionamiento. Solo se incluirá:

- *La lista de las pruebas realizadas, que debe coincidir con los casos de test que se hayan configurado en el script de simulación de Matlab.*
- *Las capturas de las formas de onda y el resumen (mostrado en la consola de Modelsim) de la simulación realizada de un máximo de 2 casos, mostrando claramente el comportamiento del circuito y que no existen errores al compararlo con el modelo de Simulink.*

Se ha incluido el ejemplo del test número 102. Debe quitarlo y añadir los casos de test que haya realizado.

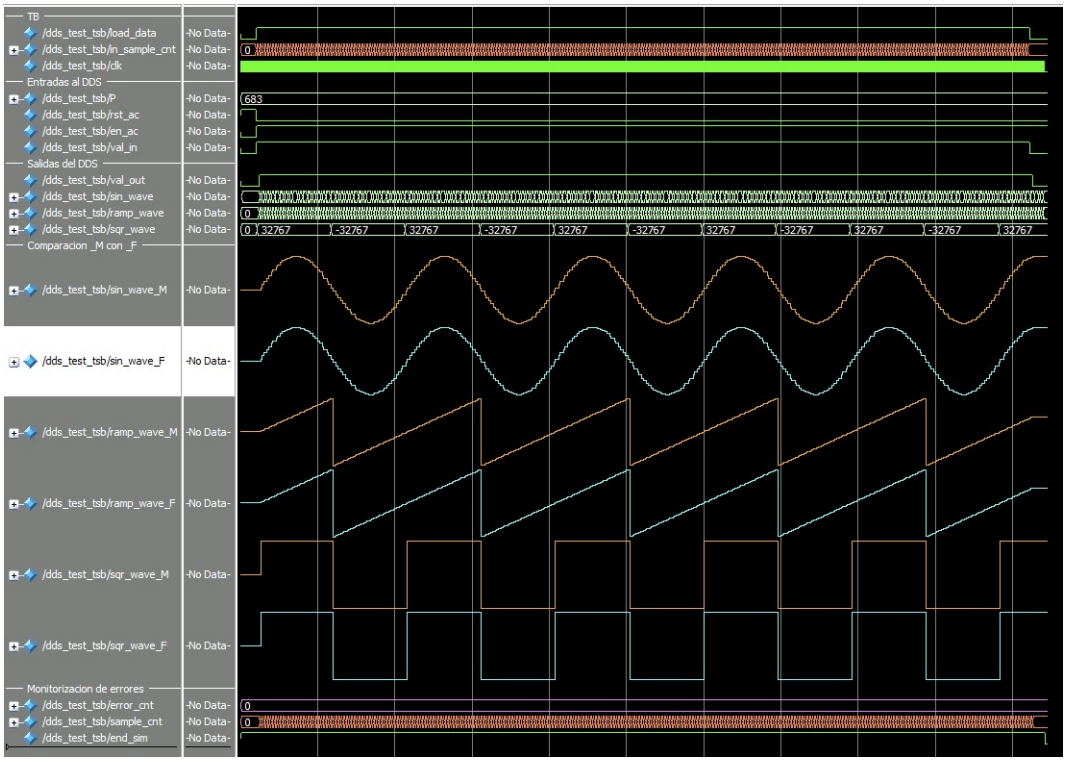
El módulo ha funcionado correctamente. Su funcionamiento se ha verificado con las siguientes pruebas:

1. Test 101: f_o = 0,1 MHz, M=27 bits, L=15 bits, W=14 bits y P=68
2. Test 102: f_o = 1 MHz, M=16 bits, L=6 bits, W=16 bits y P=683
3. Test 103: f_o = 0,0007 Hz, M=27 bits, L=15 bits, W=16 bits y P=1
4. Test 104: f_o = 48 MHz, M=27 bits, L=15 bits, W=16 bits y P=67108864
5. Test 105: f_o = 20 MHz, M=27 bits, L=15 bits, W=16 bits y P=27962027

A continuación, se describen dos de los escenarios de test enumerados anteriormente:

1.4.1. Caso 1:

A continuación se muestra el test 102: configuración del DDS con M=16 bits, L=6 bits, W=16 bits y P=683, que corresponde a una frecuencia sintetizada de 1 MHz cuando el reloj es de f_{clk} = 100 MHz. Se han simulado 193 muestras y no presenta ningún error entre las muestras simuladas y las del modelo de referencia. A continuación se muestran las diferentes señales y el resumen de la simulación mostrada en la consola de Modelsim.

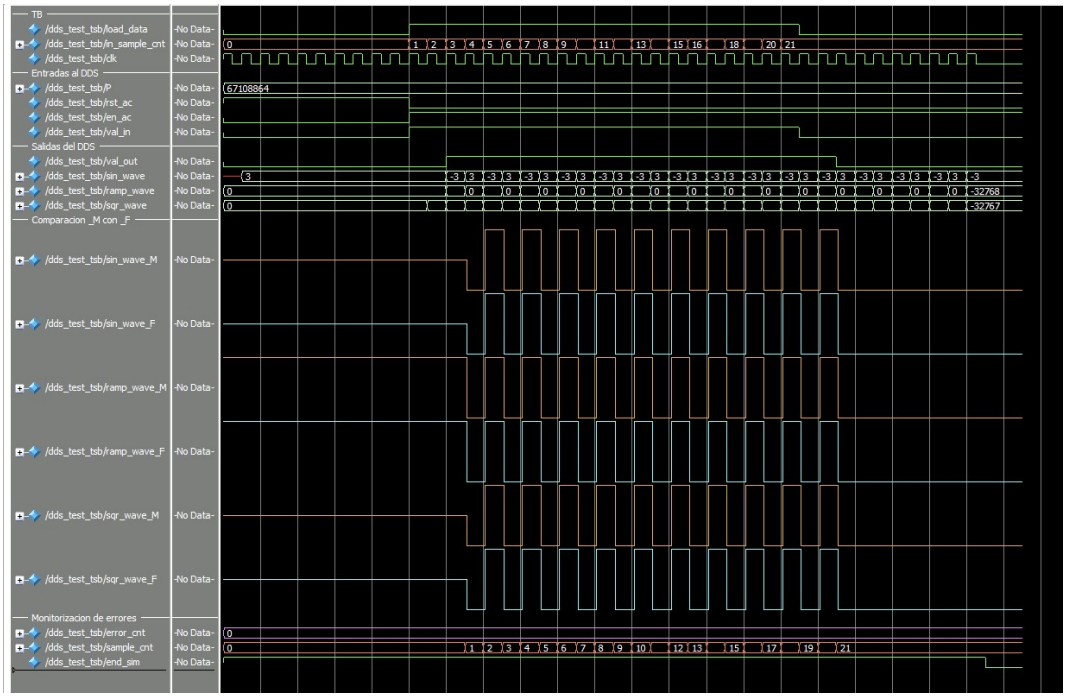



```
##########################################
# START TEST #         102
##########################################
#  Number of input samples         501
##########################################
# TEST #         102
# M =           16
# L =            6
# W =           16
# P =    683
##########################################
# Number of checked samples         501
# Number of errors           0
##########################################
```

1.4.2. Caso 2

A continuación se muestra el test 104: configuración del DDS con M=27 bits, L=15 bits, W=16 bits y P=67108864, que corresponde a una frecuencia sintetizada de 48 MHz cuando el reloj es de f_{clk} = 96 MHz. Se han simulado 21 muestras y no presenta ningún error entre las muestras simuladas y las del modelo de referencia. A continuación se

muestran las diferentes señales y el resumen de la simulación mostrada en la consola de Modelsim.




```
############################################
# START TEST #         104
############################################
#  Number of input samples          21
############################################
# TEST #         104
# M =           27
# L =           15
# W =           16
# P =  67108864
############################################
# Number of checked samples          21
# Number of errors           0
############################################
```

2. P1.2: Análisis de precisión finita de la ruta de datos del modulador de AM

ENUNCIADO 6

El modulador con precisión finita debe computar el resultado con al menos una precisión de 12 bits fraccionales. Indique los resultados del análisis de precisión finita al modular con diferentes frecuencias indicando los formatos numéricos y los bits exactos totales obtenidos. Indicar también cuál debe ser el formato numérico a aplicar para que el modulador funcione correctamente en todo el rango de frecuencias.

Tabla 5: Formatos numéricos con diferentes valores de f_{mod} y exactitud en bits

f_{mod}	Q1	Q2	Q3	Q4	Q5	Bits exactos
2 KHz	S[15,13]	S[16,13]	S[15,13]	S[16,14]	S[17,15]	12
5 KHz	S[15,13]	S[16,13]	S[15,13]	S[16,14]	S[17,15]	12
10 KHz	S[15,13]	S[16,13]	S[15,13]	S[16,14]	S[17,15]	12
20 KHz	S[15,13]	S[16,13]	S[15,13]	S[16,14]	S[17,15]	12

Tabla 6: Formatos numéricos del modulador

Q1	Q2	Q3	Q4	Q5
S[15,13]	S[16,13]	S[15,13]	S[16,14]	S[17,15]

3. Conclusiones

ENUNCIADO 7

Conclusiones de la práctica. Valore el grado de cumplimiento de los objetivos de la práctica. Haga un breve análisis de si los resultados obtenidos de área y frecuencia máxima son coherentes.

Hemos conseguido implementar un módulo de Direct Digital Synthesis, el cuál nos permite generar formas de onda de seno, rampa y cuadrada con frecuencia variable. Nuestro diseño ha sido verificado comparando con un golden model generado en Simulink y Matlab. Hemos probado diferentes casos de interés, como el caso límite de frecuencia máxima. El gasto de recursos lógicos coincide con lo esperado de nuestro diseño, y la frecuencia máxima de 196.23 MHz es coherente.

4. Ficheros entregados

ENUNCIADO 8

Lea el documento desarrollo y entrega de prácticas e indique cómo se han nombrado los ficheros que se indican en la tabla. En esta práctica esta tabla está por completar, para que sirva de referencia para las siguientes prácticas.

Los autores de la memoria confirman que han entregado los ficheros siguientes, siguiendo las pautas indicadas:

FICHERO	NOMBRE
Memoria de prácticas	Memoria_P1_G9.pdf
Entrega de los ficheros de prácticas	Proy_sim_P1_G9.zip
Proyecto Modelsim	dds_test.mpf
Fichero .do	dds_test_tsb.do
Fichero .tcl	dds_test_tsb.tcl
Proyecto Quartus	dds_test.qpf

5. ANEXO 1: Código HDL del módulo DDS_test

ENUNCIADO 9

Incluir un esquema de la división en bloques realizada para codificar el circuito y el código Verilog de los módulos de la práctica. El código debe seguir las pautas indicadas en el documento desarrollo y entrega de prácticas.

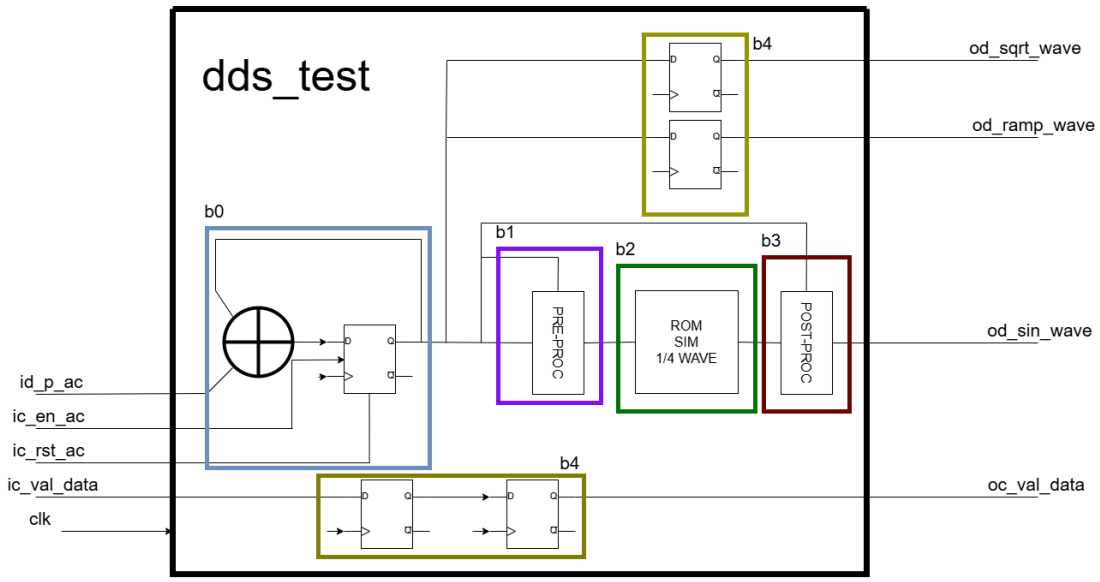


Figura 3: Diagrama de DDS test.

```
module dds_test
#(parameter M=24, // DDS accumulator wordlength
  parameter L=15, // DDS phase truncation wordlength
  parameter W=16) // DDS ROM wordlength
(
input [M-1:0] id_p_ac,        // U[M,0]
```

```
input ic_rst_ac,
input ic_en_ac,
input ic_val_data,
input clk,
output signed [W-1:0] od_sqr_wave,  // S[W,W-1]
output signed [W-1:0] od_ramp_wave,  // S[W,W-1]
output signed [W-1:0] od_sin_wave,  // S[W,W-1]
output oc_val_data
);

/* DECLARACIONES ------------------------- */

logic [M-1:0] b0_accum_r;               // U[W,0]
logic [L-1:0] b1_accum_trunc_s;         // U[L,0]
logic [L-3:0] b1_preproc_rom_addr_s;    // U[L-3,0]
logic [W-1:0] b2_rom_data_s;            // S[W,W-1]
logic [W-1:0] b3_posproc_data_s;        // S[W,W-1]
logic b3_delayed_control_r;
logic [1:0] b4_delayed_val_data_r;
logic [W-1:0] b4_sqr_wave_r;            // S[W,W-1]
logic [W-1:0] b4_ramp_wave_r;           // S[W,W-1]

/* DESCRIPCION ------------------------- */

// b0 accumulator
always_ff @(posedge clk) begin
  if(ic_rst_ac)
    b0_accum_r <= 0;
  else if(ic_en_ac)
    b0_accum_r <= b0_accum_r + id_p_ac;

end

// b1 preprocessor
assign b1_accum_trunc_s = b0_accum_r[M-1:M-L];

always_comb begin
  if(b1_accum_trunc_s[L-2] == 0) begin
    b1_preproc_rom_addr_s = b1_accum_trunc_s[L-3:0];
  end else begin
    b1_preproc_rom_addr_s = ~b1_accum_trunc_s[L-3:0];
  end
end

// b2 ROM with sine wave samples
dds_test_rom #(
    .ADDR_WIDTH(L-2),
    .DATA_WIDTH(W)
) sine_wave_rom (
    .clk(clk),
    .ic_addr(b1_preproc_rom_addr_s),
    .od_rom(b2_rom_data_s)
);
```

```
// b3 postprocessor
always_ff @(posedge clk) begin
  if(ic_rst_ac) begin
    b3_delayed_control_r <= 0;
  end else begin
    b3_delayed_control_r <= b1_accum_trunc_s[L-1];
  end
end

always_comb begin
    if(b3_delayed_control_r == 0) begin
        b3_posproc_data_s = b2_rom_data_s;
    end else begin
        b3_posproc_data_s = -b2_rom_data_s;
    end
end

// b4 Align outputs with sine wave data delay
always_ff @(posedge clk) begin
    if(ic_rst_ac) begin
    b4_ramp_wave_r <= 0;
    b4_sqr_wave_r <= 0;
    b4_delayed_val_data_r <= 0;
  end else if(ic_en_ac) begin
    b4_ramp_wave_r <= b0_accum_r[M-1:M-W];
    b4_sqr_wave_r <= b0_accum_r[M-1] == 0 ? (1 << (W - 1)) - 1 : (1 << (W - 1)) + 1;
    b4_delayed_val_data_r <= {b4_delayed_val_data_r[0], ic_val_data};
  end
end


/* ASIGNACION SALIDAS ------------------------- */

assign oc_val_data = b4_delayed_val_data_r[1];
assign od_sqr_wave = b4_sqr_wave_r;
assign od_ramp_wave = b4_ramp_wave_r;
assign od_sin_wave = b3_posproc_data_s;

endmodule


/* MEMORIA ROM PARA dds_test */
module dds_test_rom
#(parameter ADDR_WIDTH=13, // Address wordlength
  parameter DATA_WIDTH=14) // ROM output wordlength
(
  input [ADDR_WIDTH-1:0] ic_addr,      // U[ADDR_WIDTH,0]
  input clk,
  output signed [DATA_WIDTH-1:0] od_rom    // S[DATA_WIDTH,DATA_WIDTH-1]
);

/* DECLARACIONES ------------------------- */
reg signed [DATA_WIDTH-1:0] rom [0:2**ADDR_WIDTH-1];
```

```
logic [DATA_WIDTH-1:0] b0_rom_r;

/* DESCRIPCION ------------------------- */

// ROM data
initial
  if ((DATA_WIDTH == 14)&(ADDR_WIDTH == 13))
      $readmemb("../src/rom_dds_L15_W14.txt", rom);
  else  if ((DATA_WIDTH == 16)&(ADDR_WIDTH == 13))
      $readmemb("../src/rom_dds_L15_W16.txt", rom);
  else  if ((DATA_WIDTH == 16)&(ADDR_WIDTH == 4))
      $readmemb("../src/rom_dds_L6_W16.txt", rom);


// Read synchronous ROM
always_ff@ (posedge clk)
      b0_rom_r <= rom[ic_addr];


/* ASIGNACION SALIDAS ------------------------- */
assign od_rom = b0_rom_r;

endmodule
```