

## Capítulo 5

# Simulación y verificación de las descripciones

Luego de describir un circuito digital, éste debe ser simulado para verificar y validar su comportamiento. Se realizaron simulaciones para verificar el funcionamiento de la descripción funcional algorítmica y de las arquitecturas particulares. Las simulaciones realizadas son de índole funcional, lo que permitió verificar el funcionamiento correcto de cada descripción. Se modificaron dos parámetros de interés: el ancho de palabra y el número de iteraciones. En las simulaciones no se tomaron en cuenta retrasos ni tiempos de los componentes, ya que los retrasos adquieren sentido cuando se tiene en consideración una plataforma específica.

### 5.1 Herramientas de simulación

Para realizar la simulación de los diseños, se empleó el simulador VeriBest VBVHDL 99.0 que forma parte de la herramienta de desarrollo junto con el compilador de VHDL.

La simulación de un diseño, se lleva a cabo estableciendo la entidad que se encuentra en el tope de la jerarquía del diseño que se desea simular. Esta entidad, denominada Entidad Raíz (Root Entity), usualmente será el banco de pruebas que se utiliza para llevar a cabo las verificaciones.

El código VHDL compilado, puede ser ejecutado y depurado con el uso del debugger. Para realizar una simulación se debe ejecutar el simulador y seguidamente se debe establecer el tiempo de duración de la simulación.

Asimismo el ambiente de trabajo incluye un Visor de Ondas (Waveform Viewer) cuyo aspecto se muestra en la figura 5.1. Este visor permite monitorear los valores de las señales luego de la simulación. Se pueden abrir tantas ventanas como se desee para monitorear diversas partes de un diseño. Las señales se pueden incluir de a una o por componentes (figura 5.2).

Los valores de las señales se pueden analizar una vez que la simulación ha terminado, al haberse alcanzado el tiempo fijado previamente. El simulador almacena los valores de la simulación completa en archivos temporales en el disco rígido. Este último aspecto puede resultar problemático para simulaciones de larga duración y diseños que poseen una gran cantidad de señales.

El visor de ondas únicamente tiene sentido para simulaciones en donde el diseño emplea señales. Los diseños algorítmicos hacen uso de variables que son elementos del lenguaje de alto nivel y no pueden ser visualizadas. Sin embargo para analizar diseños funcionales secuenciales el simulador cuenta con un debugger que es idéntico al de cualquier lenguaje de programación

convencional. Este debugger permite ubicar puntos de parada (breakpoints) y monitorear el contenido de variables.

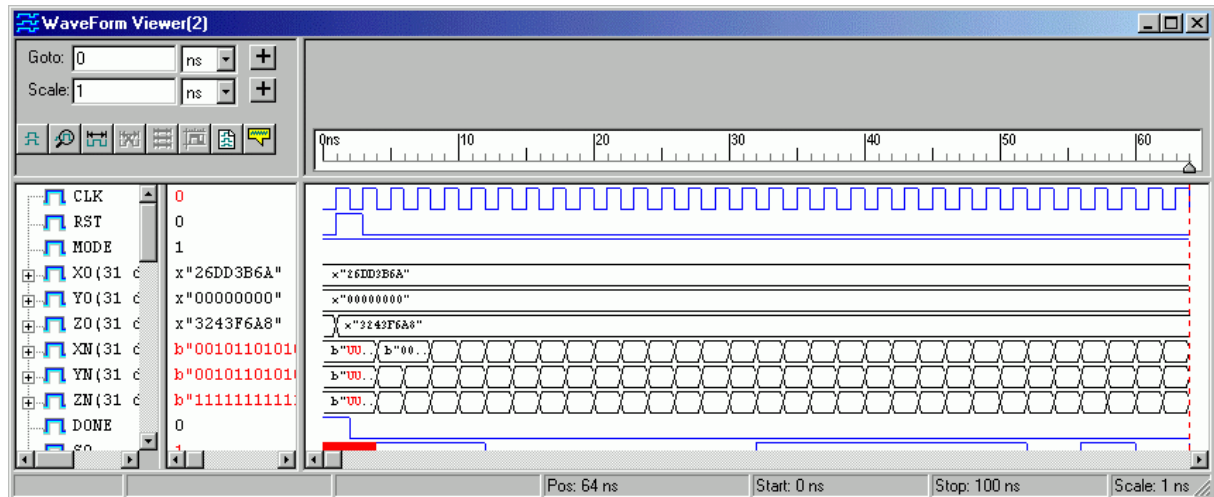


Figura 5.1

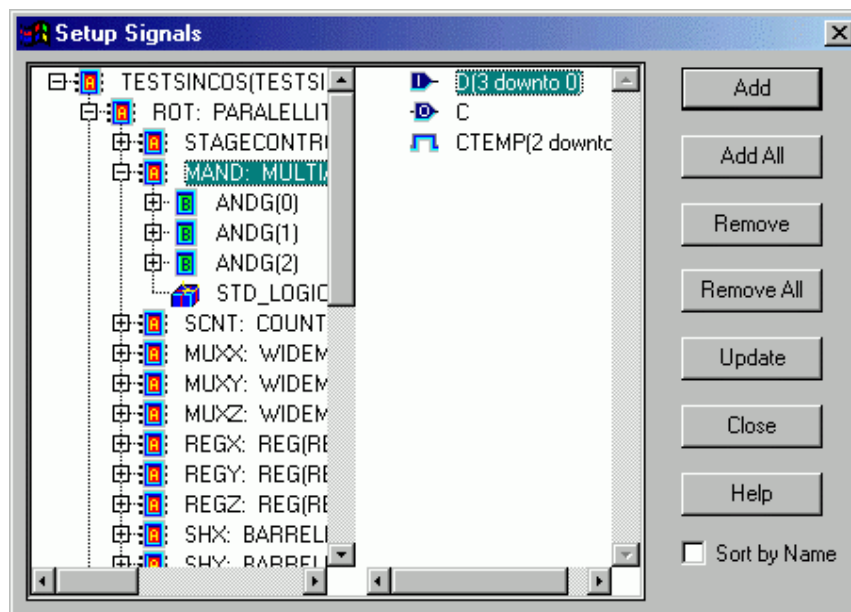


Figura 5.2

Las simulaciones de la descripción funcional algorítmica realizaron sobre una computadora basada en un procesador Intel Pentium de 200Mhz con 32Mb de memoria RAM y 2Gb de espacio libre en disco rígido. Para simular las descripciones de las arquitecturas particulares, se recurrió a una computadora basada en un procesador Intel Pentium III de 800Mhz con 128Mb de memoria RAM y 35Gb de espacio libre en disco rígido. Se optó por cambiar de sistema debido a la capacidad de memoria y procesamiento que requería el simulador para funcionar adecuadamente.

## 5.2 Valores para la simulación

La simulación del seno y del coseno se realizó para los ángulos  $\theta$  entre  $-\frac{\pi}{2}$  y  $\frac{\pi}{2}$ , rango de convergencia del algoritmo CORDIC, y para los ángulos  $-\frac{\pi}{2}, -\frac{\pi}{3}, -\frac{\pi}{4}, -\frac{\pi}{6}, 0, \frac{\pi}{6}, \frac{\pi}{4}, \frac{\pi}{3}, \frac{\pi}{2}$ .

Para el arcotangente, el rango para las descripciones que utilizan aritmética de punto fijo, se limitó por un lado a los valores de la tangente entre  $\operatorname{tg}\left(-\frac{4\pi}{9}\right)$  y  $\operatorname{tg}\left(\frac{4\pi}{9}\right)$ . Por otro lado se realizó la simulación en todo el rango de convergencia del algoritmo, como se explicó en la sección 4.3.1. La función recíproca  $f(x) = \frac{1}{x}$  no se describió en VHDL, los valores adecuados fueron suministrados por el banco de pruebas y la biblioteca matemática del lenguaje.

El incremento de iteración que se escogió para el rango es de  $1^\circ$  (un minuto angular) que expresado en radianes es  $\frac{\pi}{10800}$ .

Las simulaciones de la descripción algorítmica no llevaron gran cantidad de tiempo, sin embargo la simulación de cada arquitectura particular demandó en promedio entre 3 y 4 horas. El tamaño de los archivos temporales para las arquitecturas, almacenados durante la simulación, fue de aproximadamente 20Gb.

No se describió un banco de pruebas para suministrar todas las posibles entradas de acuerdo al ancho de palabra, debido al tiempo que demandaría el cálculo. Para las entradas con ancho de palabra de 16 bits y 16 iteraciones, por ejemplo, el cálculo de 10800 valores demandó aproximadamente 3 horas.

Los valores para las tres descripciones se compararon con los calculados por MATLAB™ [Anexo C]. La comparación se realizó calculando el error absoluto  $E$  que se obtuvo entre ambos métodos. Para calcular el error se utilizó la siguiente fórmula:

$$E(x) = |V_{CORDIC}(x) - V_{MATLAB}(x)|$$

El error absoluto puede expresarse en bits y en la práctica puede utilizarse para decidir cuantos bits se consideran exactos para la representación de los resultados. Se emplea la siguiente fórmula:

$$D_2(x) = \lceil -\log_2 E(x) \rceil - 1$$

En ambos casos,  $V(x)$  representa el valor calculado por el algoritmo.

Las simulaciones se llevaron a cabo modificando el número de iteraciones entre 4 y un máximo, que para las arquitecturas particulares está determinado por el ancho de palabra y para la descripción funcional algorítmica se estableció en 32. Para cada número de iteraciones se realizaron las simulaciones correspondientes dentro del rango de convergencia. El número de iteraciones se incrementó de a dos.

### 5.3 Simulación de la descripción funcional algorítmica

La simulación de la descripción funcional algorítmica se llevó a cabo para validar el funcionamiento del algoritmo CORDIC. Los resultados se compararon con los valores que se obtuvieron al evaluar dichas funciones con MATLAB™. La descripción funcional opera con el tipo REAL de VHDL que es una representación en punto flotante basada en el estándar IEEE 754 de doble precisión.

A modo de ejemplo, se siguen los pasos que efectúa el algoritmo para calcular el seno y el coseno de  $\frac{\pi}{4} \cong 0,78539$  y se exponen los valores que adquieren los distintos parámetros del mismo.

El número de dígitos fraccionarios se restringió a cinco ya que el objetivo del ejemplo es únicamente el de ilustrar el funcionamiento del algoritmo. Al final del cálculo, los valores del coseno y del seno, se obtienen de las variables X e Y respectivamente.

i	X	Y	Z	$2^{-i}$	$\arctg(2^{-i})$	signo(Z)
0	0.60725	0.00000	0.78539	1.00000	0.78539	+
1	0.60725	0.60725	-0.00000	0.50000	0.46364	-
2	0.91087	0.30362	0.46364	0.25000	0.24497	+
3	0.83497	0.53134	0.21866	0.12500	0.12435	+
4	0.76855	0.63571	0.09431	0.06250	0.06241	+
5	0.72882	0.68375	0.03189	0.03125	0.03123	+
6	0.70745	0.70652	0.00065	0.01562	0.01562	+
7	0.69641	0.71758	-0.01496	0.00781	0.00781	-
8	0.70202	0.71214	-0.00715	0.00390	0.00390	-
9	0.70480	0.70939	-0.00324	0.00195	0.00195	-
10	0.70618	0.70802	-0.00129	0.00097	0.00097	-
11	0.70688	0.70733	-0.00032	0.00048	0.00048	-
12	0.70722	0.70698	0.00016	0.00024	0.00024	+
13	0.70705	0.70716	-0.00007	0.00012	0.00012	-
14	0.70713	0.70707	0.00004	0.00006	0.00006	+
15	0.70709	0.70711	-0.00001	0.00003	0.00003	-
16	<b>0.70711</b>	<b>0.70709</b>	0.00001	0.00001	0.00001	+

A continuación se muestran los resultados de la simulación para los valores  $-\frac{\pi}{2}, -\frac{\pi}{3}, -\frac{\pi}{4}, -\frac{\pi}{6}, 0, \frac{\pi}{6}, \frac{\pi}{4}, \frac{\pi}{3}, \frac{\pi}{2}$  con 32 iteraciones en las tablas 5.1, 5.2 y 5.3 respectivamente. En las tablas también se muestra el valor obtenido por MATLAB™ a modo de referencia.

Angulo	Seno MATLAB	Seno CORDIC
$-\frac{\pi}{2}$	-1	-1
$-\frac{\pi}{3}$	-8.660254037844386e-001	-8.660254038110794e-001
$-\frac{\pi}{4}$	-7.071067811865475e-001	-7.071067811182992e-001
$-\frac{\pi}{6}$	-4.999999999999999e-001	-4.99999999538575e-001
0	0	1.295764953995623e-010
$\frac{\pi}{6}$	4.999999999999999e-001	4.99999999538575e-001
$\frac{\pi}{4}$	7.071067811865475e-001	7.071067812547968e-001
$\frac{\pi}{3}$	8.660254037844386e-001	8.660254038110794e-001
$\frac{\pi}{2}$	1	1

**Tabla 5.1**

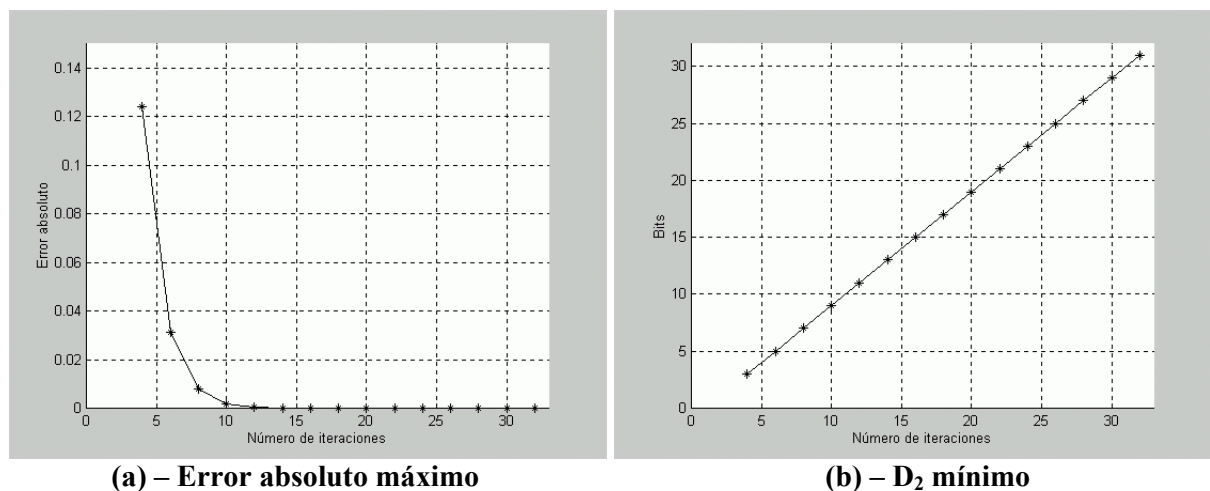
Angulo	Coseno MATLAB	Coseno CORDIC
$-\frac{\pi}{2}$	0	1.295764953995623e-010
$-\frac{\pi}{3}$	5.000000000000001e-001	4.99999999538575e-001
$-\frac{\pi}{4}$	7.071067811865476e-001	7.071067812547968e-001
$-\frac{\pi}{6}$	8.660254037844387e-001	8.660254038110794e-001
0	1	1
$\frac{\pi}{6}$	8.660254037844387e-001	8.660254038110794e-001
$\frac{\pi}{4}$	7.071067811865476e-001	7.071067811182992e-001
$\frac{\pi}{3}$	5.000000000000001e-001	4.99999999538575e-001
$\frac{\pi}{2}$	0	1.295764953995623e-010

**Tabla 5.2**

Tangente	Arcotangente MATLAB	Arcotangente CORDIC
$\text{tg}\left(-\frac{\pi}{2}\right)$	-1.570796326794896	-1.570796326665320
$\text{tg}\left(-\frac{\pi}{3}\right)$	-1.047197551196598	-1.047197551249879
$\text{tg}\left(-\frac{\pi}{4}\right)$	-7.853981633974483e-001	-7.853981633009298e-001
$\text{tg}\left(-\frac{\pi}{6}\right)$	-5.235987755982988e-001	-5.235987755450179e-001
$\text{tg}(0)$	0	1.295765189457968e-010
$\text{tg}\left(\frac{\pi}{6}\right)$	5.235987755982988e-001	5.235987755450179e-001
$\text{tg}\left(\frac{\pi}{4}\right)$	7.853981633974483e-001	7.853981633009298e-001
$\text{tg}\left(\frac{\pi}{3}\right)$	1.047197551196598	1.047197551249879
$\text{tg}\left(\frac{\pi}{2}\right)$	1.570796326794896	1.570796326665320

**Tabla 5.3**

Se calcularon los valores del seno, del coseno y del arcotangente en el rango de convergencia del algoritmo, modificando el número de iteraciones entre 4 y 32. A partir de los valores obtenidos mediante las simulaciones y MATLAB™, se calculó el error absoluto máximo. Con dicho error se obtuvo la mínima cantidad de bits considerados exactos, según la fórmula  $D_2$ . Los valores obtenidos se muestran en las tablas 5.4 y 5.5, y se grafican en la figura 5.3. La figura 5.3 a) corresponde a la gráfica del error absoluto máximo y la figura 5.3 b) a la cantidad mínima de bits válidos para cada iteración. Los gráficos del error para las tres funciones son similares por lo que se expone solo uno, sin embargo existen diferencias que pueden observarse en la tabla 5.4. En el caso de la cantidad mínima de dígitos binarios considerados exactos los valores son idénticos para las tres funciones simuladas.


**Figura 5.3**

Iteraciones	Error absoluto máximo		
	Seno	Coseno	Arcotangente
4	1.239941434116256e-001	1.239941434117357e-001	1.243323987289236e-001
6	3.123814548303316e-002	3.123814548316328e-002	3.123942535351132e-002
8	7.802869457692359e-003	7.802869457822481e-003	7.803355672933332e-003
10	1.943815911317360e-003	1.943815911447482e-003	1.950192443141596e-003
12	4.801867943289434e-004	4.801867944251720e-004	4.865316636393224e-004
14	1.217712861756731e-004	1.217712860590511e-004	1.220351082866778e-004
16	3.049289234331211e-005	3.049289221112618e-005	3.049398127026766e-005
18	7.605526813447838e-006	7.605526681261909e-006	7.627125881315422e-006
20	1.897498003292086e-006	1.897498132005099e-006	1.906345085678218e-006
22	4.737062145668269e-007	4.737060940937510e-007	4.767281169826276e-007
24	1.185509823607711e-007	1.185508512833306e-007	1.191542309264548e-007
26	2.968034067130088e-008	2.968021136154352e-008	2.977507684853720e-008
28	7.444936066941521e-009	7.445063551769770e-009	7.448033967349943e-009
30	1.858476015442756e-009	1.858603500271006e-009	1.862529797058698e-009
32	4.624036500922024e-010	4.622801516585006e-010	4.656410812486910e-010

**Tabla 5.4**

Iteraciones	D <sub>2</sub> mínimo
	Bits
4	3
6	5
8	7
10	9
12	11
14	13
16	15
18	17
20	19
22	21
24	23
26	25
28	27
30	29
32	31

**Tabla 5.5**

Se puede observar en la figura 5.3 a) y en la tabla 5.4 que el error absoluto disminuye a medida que aumenta el número de iteraciones. En la figura 5.3 b) y en la tabla 5.5 se puede ver como se incrementa el mínimo en bits exactos.

## 5.4 Simulación de las arquitecturas particulares

### 5.4.1 Cálculo de valores

La simulación de las descripciones que utilizan aritmética de punto fijo se realizó con un ancho de palabra de 16 y 32 bits. Los valores obtenidos para el seno y el coseno tanto en la arquitectura bit-paralela desplegada como en la iterativa fueron iguales, ya que si bien se trata de dos arquitecturas distintas, la funcionalidad es la misma para ambas.

La exactitud de los resultados proporcionados por el algoritmo CORDIC no depende únicamente del ancho de palabra de las componentes  $X$ ,  $Y$  y  $Z$ , sino también del número de iteraciones que ejecuta el algoritmo. Se observó que el número de iteraciones está limitado por el ancho de palabra, es decir que con un ancho de  $n$  bits se obtienen  $n$  iteraciones útiles. Este hecho se debe a la multiplicación del factor  $2^{-i}$  por las componentes  $X$  e  $Y$ , en donde  $i$  representa el número de iteración. Esta multiplicación equivale a desplazar las componentes hacia la derecha  $i$  veces. Entonces, cuando  $i \geq n$ , el resultado del desplazamiento para un número positivo valdrá cero. Para la componente  $Z$  o acumulador angular ocurre algo similar con el valor de  $\arctg(2^{-i})$ . A modo de ejemplo se muestra la tabla de arcotangentes para un ancho de palabra de 32 bits en VHDL. Cuando  $i \geq 32$ ,  $\arctg(2^{-i})$  vale cero.

```
constant arctanLUT:STDVectorW(0 to 31):=(
    "11001001000011111101101010100010",
    "01110110101100011001110000010101",
    "00111110101101101110101111110010",
    "00011111110101011011101010011011",
    "0000111111110101010110111011100",
    "000001111111111010101010101111",
    "000000111111111110101010101011",
    "000000011111111111101010101011",
    "000000001111111111111110101011",
    "00000000011111111111111111101011",
    "00000000001111111111111111110101",
    "00000000000111111111111111111111",
    "00000000000011111111111111111111",
    "00000000000001111111111111111111",
    "00000000000000111111111111111111",
    "00000000000000011111111111111111",
    "00000000000000001111111111111111",
    "00000000000000000111111111111111",
    "00000000000000000011111111111111",
    "00000000000000000001111111111111",
    "00000000000000000000111111111111",
    "00000000000000000000011111111111",
    "00000000000000000000001111111111",
    "00000000000000000000000111111111",
    "00000000000000000000000011111111",
    "00000000000000000000000001111111",
    "00000000000000000000000000111111",
    "00000000000000000000000000011111",
    "00000000000000000000000000001111",
    "00000000000000000000000000000111",
    "00000000000000000000000000000011",
    "00000000000000000000000000000001",
    "000000000000000000000000000000001"
);
```

Con el formato numérico explicado en la sección 4.3.1 se utiliza todo el ancho de palabra de la tabla de arcotangentes. Si se hubiesen reservado lugares para la parte entera dentro de la tabla de



búsqueda, se perderían dígitos menos significativos, limitando aún mas el número de iteraciones útiles.

Para las descripciones particulares también se calcularon los valores de los ángulos  $-\frac{\pi}{2}, -\frac{\pi}{3}, -\frac{\pi}{4}, -\frac{\pi}{6}, 0, \frac{\pi}{6}, \frac{\pi}{4}, \frac{\pi}{3}, \frac{\pi}{2}$ . Los valores se muestran en las tablas 5.6, 5.7 y 5.8 para 16 y 32 bits efectuando 16 y 32 iteraciones respectivamente. Se los convirtió a punto flotante con MATLAB™ a fin de obtener un resultado legible. Las arcotangentes para  $\operatorname{tg}\left(-\frac{\pi}{2}\right)$  y  $\operatorname{tg}\left(\frac{\pi}{2}\right)$  se calcularon sustituyendo  $y_0 = \pm 1$ ,  $x_0 = 0$  y  $z_0 = 0$  en la ecuación  $z_n = z_0 + \operatorname{arctg}\left(\frac{y_0}{x_0}\right)$ , como se explicó en la sección 4.3.1.

Angulo	Seno MATLAB	Seno CORDIC (16 bits)	Seno CORDIC (32 bits)
$-\frac{\pi}{2}$	-1	-1	-1
$-\frac{\pi}{3}$	-8.660254037844386e-001	-8.660888671875000e-001	-8.660254040732980e-001
$-\frac{\pi}{4}$	-7.071067811865475e-001	-7.070312500000000e-001	-7.071067783981562e-001
$-\frac{\pi}{6}$	-4.999999999999999e-001	-5.000000000000000e-001	-4.999999981373549e-001
0	0	0	0
$\frac{\pi}{6}$	4.999999999999999e-001	4.999389648437500e-001	4.999999981373549e-001
$\frac{\pi}{4}$	7.071067811865475e-001	7.070312500000000e-001	7.071067774668336e-001
$\frac{\pi}{3}$	8.660254037844386e-001	8.660888671875000e-001	8.660254031419754e-001
$\frac{\pi}{2}$	1	9.999389648437500e-001	9.99999990686774e-001

**Tabla 5.6**

Angulo	Coseno MATLAB	Coseno CORDIC (16 bits)	Coseno CORDIC (32 bits)
$-\frac{\pi}{2}$	0	-6.103515625000000e-005	0
$-\frac{\pi}{3}$	5.000000000000001e-001	5.000000000000000e-001	4.999999962747097e-001
$-\frac{\pi}{4}$	7.071067811865476e-001	7.071533203125000e-001	7.071067811921239e-001
$-\frac{\pi}{6}$	8.660254037844387e-001	8.660278320312500e-001	8.660254050046206e-001
0	1	9.999389648437500e-001	9.99999990686774e-001
$\frac{\pi}{6}$	8.660254037844387e-001	8.660888671875000e-001	8.660254031419754e-001
$\frac{\pi}{4}$	7.071067811865476e-001	7.070922851562500e-001	7.071067811921239e-001
$\frac{\pi}{3}$	5.000000000000001e-001	4.999389648437500e-001	4.999999981373549e-001
$\frac{\pi}{2}$	0	0	9.313225746154784e-010

**Tabla 5.7**

Tangente	Arcotangente MATLAB	Arcotangente CORDIC (16 bits)	Arcotangente CORDIC (32 bits)
$\text{tg}\left(-\frac{\pi}{2}\right)$	-1.570796326794896	-1.570800781250000	-1.570796329528093
$\text{tg}\left(-\frac{\pi}{3}\right)$	-1.047197551196598	-1.047241210937500	-1.047197550535202
$\text{tg}\left(-\frac{\pi}{4}\right)$	-7.853981633974483e-001	-7.854003906250000e-001	-7.853981647640467e-001
$\text{tg}\left(-\frac{\pi}{6}\right)$	-5.235987755982988e-001	-5.235957031250000e-001	-5.235987771302462e-001
$\text{tg}(0)$	0	0	-1.862645149230957e-009
$\text{tg}\left(\frac{\pi}{6}\right)$	5.235987755982988e-001	5.234375000000000e-001	5.235987752676010e-001
$\text{tg}\left(\frac{\pi}{4}\right)$	7.853981633974483e-001	7.852783203125000e-001	7.853981629014015e-001
$\text{tg}\left(\frac{\pi}{3}\right)$	1.047197551196598	1.047119140625000	1.047197548672557
$\text{tg}\left(\frac{\pi}{2}\right)$	1.570796326794896	1.570678710937500	1.570796327665448

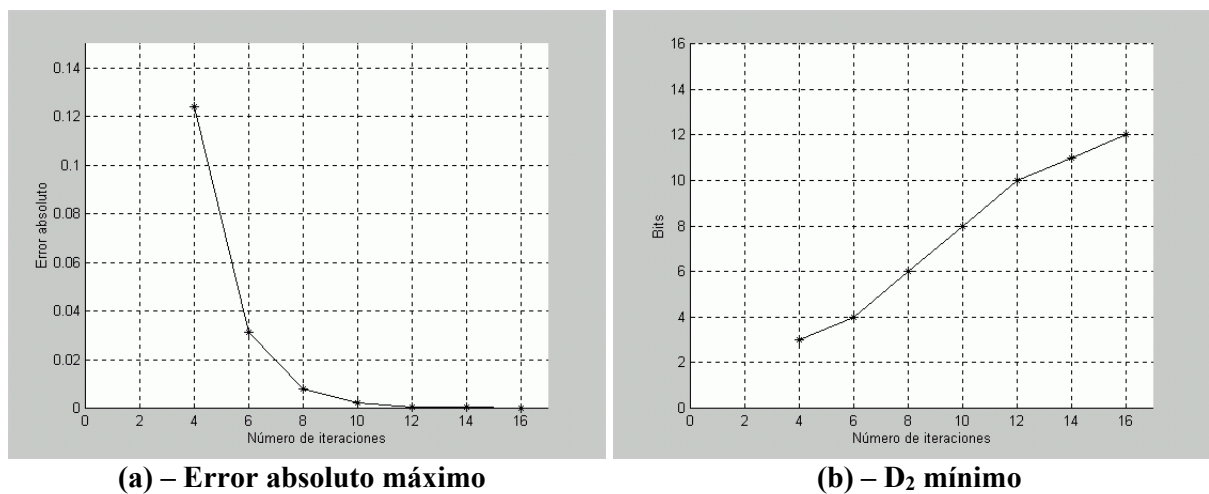
**Tabla 5.8**

Para las descripciones particulares se calculó el seno, el coseno y el arcotangente dentro de los rangos establecidos en la sección 5.2. El arcotangente, se calculó además para la tangente entre  $\text{tg}\left(-\frac{\pi}{2}\right)$  y  $\text{tg}\left(\frac{\pi}{2}\right)$  utilizando la función recíproca provista en el banco de pruebas. De ésta manera se pudo reducir a 3 la cantidad de bits correspondientes a la parte entera y se pudo cubrir todo el rango de convergencia del algoritmo CORDIC.

En las figuras 5.4 y 5.5 se exponen el error absoluto (figuras 5.4 y 5.5 a)) y los bits de acuerdo a la fórmula  $D_2$  (figuras 5.4 y 5.5 b)) para el seno y el coseno, con un ancho de palabra de 16 y 32 bits respectivamente.

En las tablas 5.9, 5.10, 5.11 y 5.12 se muestran a modo de resumen los extremos de los resultados obtenidos a partir de las descripciones particulares, junto con los de la descripción funcional algorítmica. Las tablas de valores detallados y los gráficos restantes se han reservado para el anexo E.

Se puede observar que el error absoluto máximo disminuye y la cantidad mínima de bits considerados exactos crece cuando aumenta el número de iteraciones que efectúa el algoritmo de forma parecida a lo que ocurría con la descripción funcional algorítmica. De acuerdo a los resultados de las simulaciones, se puede inferir que cada iteración adicional incrementa la exactitud del resultado en 1 bit aproximadamente.



**Figura 5.4 – Seno / Coseno (16 bits)**

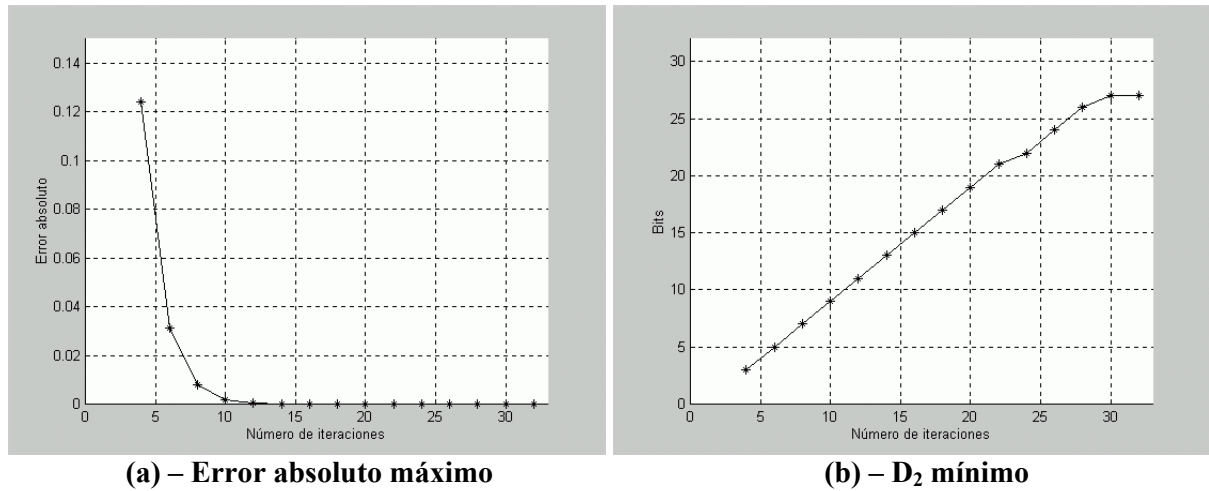


Figura 5.5 – Seno / Coseno (32 bits)

Seno para ángulos entre $-\frac{\pi}{2}$ y $\frac{\pi}{2}$			
Ancho	Iteraciones	E (máximo)	$D_2$ (mínimo)
Tipo REAL	4	1.239941434116256e-001	3
	32	4.624036500922024e-010	31
16 bits	4	1.242654148457032e-001	3
	16	1.958996423664927e-004	12
32 bits	4	1.239941431395144e-001	3
	32	5.156055915556124e-009	27

Tabla 5.9

Coseno para ángulos entre $-\frac{\pi}{2}$ y $\frac{\pi}{2}$			
Ancho	Iteraciones	E (máximo)	$D_2$ (mínimo)
Tipo REAL	4	1.239941434117357e-001	3
	32	4.622801516585006e-010	31
16 bits	4	1.240364158710175e-001	3
	16	2.309664794707678e-004	12
32 bits	4	1.239941440706783e-001	3
	32	6.584441936130503e-009	27

Tabla 5.10

Arcotangente para la tangente entre $\operatorname{tg}\left(-\frac{\pi}{2}\right)$ y $\operatorname{tg}\left(\frac{\pi}{2}\right)$			
Ancho	Iteraciones	E (máximo)	D <sub>2</sub> (mínimo)
Tipo REAL	4	1.243323987289236e-001	3
	32	4.656410812486910e-010	31
16 bits	4	1.244063634178296e-001	3
	16	3.053631790548294e-004	11
32 bits	4	1.243324004358084e-001	3
	32	5.184624507492685e-009	27

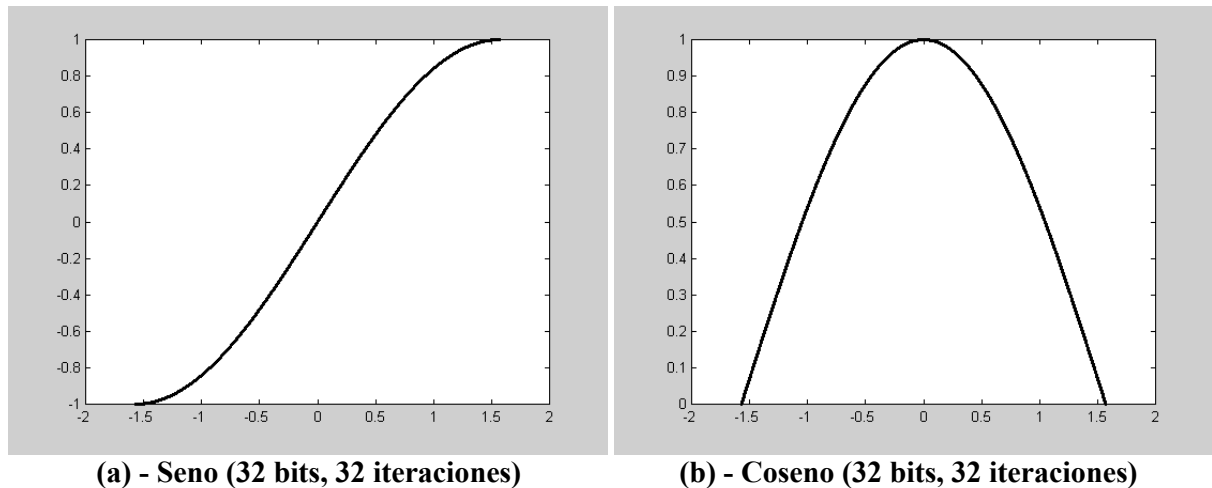
**Tabla 5.11**

Arcotangente para la tangente entre $\operatorname{tg}\left(-\frac{4\pi}{9}\right)$ y $\operatorname{tg}\left(\frac{4\pi}{9}\right)$			
Ancho	Iteraciones	E (máximo)	D <sub>2</sub> (mínimo)
16 bits	4	1.250465874394296e-001	2
	16	9.284920377063966e-004	10
32 bits	4	1.243324013065256e-001	3
	32	1.486753375967708e-008	26

**Tabla 5.12**

Como se observa en las tablas anteriores el error absoluto máximo disminuye a medida que aumentan las iteraciones. Los valores para el error son aproximados, un mayor detalle de los valores se expone en el anexo E.

Por último, en la figura 5.6 se muestra a modo de ejemplo la gráfica del seno y del coseno dentro del rango de convergencia con un ancho de palabra de 32 bits, para 32, 6 y 4 iteraciones. Como puede observarse, con un número bajo de iteraciones, aumenta la repetición de valores.


**Figura 5.6**

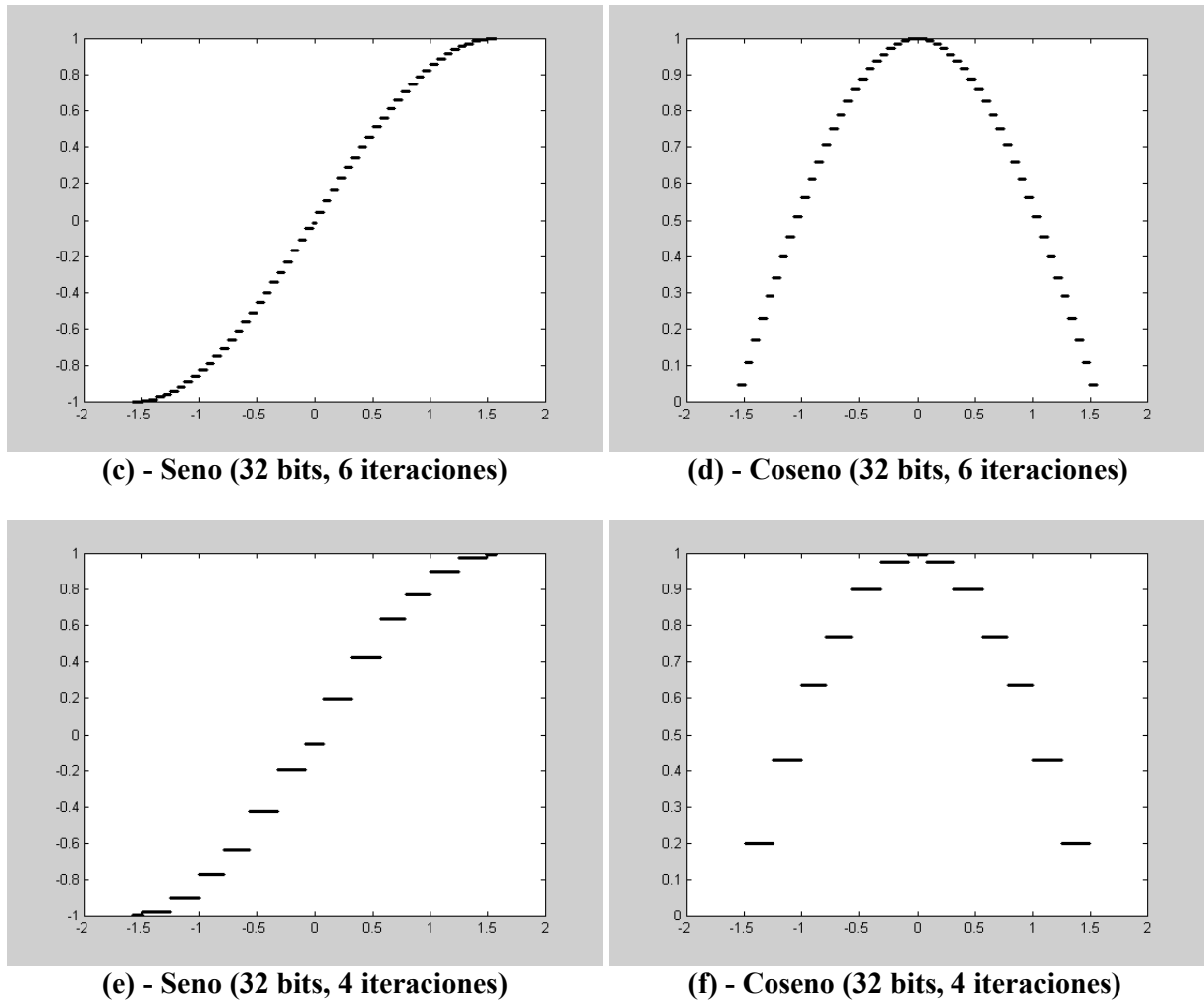


Figura 5.6

### 5.4.2 Visualización de ondas

El visor de ondas es una herramienta del simulador de VHDL que resulta especialmente útil durante el diseño. Su función es permitir la visualización y monitorización de señales. La visualización que se expone a continuación es una simulación típica de la descripción de la arquitectura bit-paralela iterativa. Muestra los cambios que se producen en las señales conforme avanza el tiempo de simulación. Se ejemplifica con el cálculo de las funciones seno y coseno para el ángulo  $\frac{\pi}{4}$  con un ancho de palabra de 32 bits y para 32 iteraciones. El tiempo de simulación se establece en la descripción del banco de pruebas. En el caso de la simulación funcional no es importante la unidad de tiempo que se utilice. El tiempo de simulación adquiere importancia en las simulaciones digitales, porque es ahí en donde se toman en consideración los retrasos y tiempos de los componentes. Sin embargo se debe establecer un tiempo arbitrario para poder obtener los resultados adecuadamente y provocar el avance del tiempo en el simulador. Para la simulación que se muestra a continuación el tiempo está determinado por el reloj, y se estableció en 1 ns.

