SKRIPSI

PERANCANGAN DAN PEMBUATAN GENERATOR SINYAL UNTUK SISTEM PENGUJI RANGKAIAN ELEKTRONIK BERBASIS MICROKONTROLLER AT89C51

Disusun Oleh :

DIDI AFFANDI

NIM. 98.17.006

JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK ELEKTRONIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG

2010

LEMBAR PERSETUJUAN

PERANCANGAN DAN PEMBUATAN SISTEM PENGUJI RANGKAIAN ELEKTRONIK PENGUAT SINYAL AUDIO BERBASIS PC

SKRIPSI

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Teknik Pada Jurusan Teknik Elektro S-1 Konsentrasi Teknik Elektronika

Disusun Oleh :

DIDI AFFANDI

NIM : 98.17.006

Diperiksa dan Disetujui

Dosen Pembimbing I

Ir. F. Yudi Limpraptono, MT
NIP.Y.1039500274

Mengetahui
Ketua Jurusan Teknik Elektro S-1

Ir. F. Yudi Limpraptono, MT
NIP.Y.1039500274

JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK ELEKTRONIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL
MALANG
2010

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1
Jl. Raya Karanglo Km.2
Malang

BERITA ACARA UJIAN SKRIPSI

Nama	: DIDI AFFANDI
N.I.M	: 98.17.006
Jurusan	: TEKNIK ELEKTRO S-1
Konsentrasi	: TEKNIK ELEKTRONIKA S-1
Judul Skripsi	: PERANCANGAN DAN PEMBUATAN GENERATOR SINYAL UNTUK SISTEM PENGUJI RANGKAIAN ELEKTRONIK BERBASIS MICROKONTROLLER AT89C51

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :

Hari	: Senin
Tanggal	: 23 Maret 2009
Dengan Nilai	: 74 (B+)

Panitia Majelis Penguji :

Ketua

Ir. Sidik Noertjahjono, MT
NIP.Y. 1028700163

Sekretaris

Ir. F. Yudi Limpraptono, MT
NIP.Y. 1039500274

Anggota Penguji :

Penguji Pertama

Joseph Dedy Irawan, ST, MT
NIP. 132315178

Penguji Kedua

Setiawardhi, ST
NIP.Y. 1039700309

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1
Jl. Raya Karanglo Km.2
Malang

FOMULIR PERBAIKAN SKRIPSI

Nama : Didi Affandi
NIM : 98.17.006
Masa Bimbingan : 20 Desember 2008 – 20 Juni 2009
Judul Skripsi : PERANCANGAN DAN PEMBUATAN GENERATOR SINYAL UNTUK SISTEM PENGUJI RANGKAIAN ELEKTRONIK BERBASIS MIKROKONTROLLER AT89C51

NO.	TANGGAL	URAIAN	PARAF
1	23 MARET 2009	DESAIN TABEL, DARI MANA?	
2	23 MARET 2009	FREKWENSI YANG DIINGINKAN (DIHITUNG)	
3	23 MARET 2009	ERROR (DIBANDINGKAN DENGAN BAB III)	

Disetujui,
Penguji I

Joseph Dedy Irawan, ST, MT
NIP. 132315178

Mengetahui,
Dosen Pembimbing

Ir. F. Yudi Limpraptono, MT
NIP.Y. 1039500274

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat ALLAH SWT, karena berkat Rahmat dan Hidayah-Nya skripsi dan tugas akhir ini dapat selesai dengan baik.

Pada awal pelaksanaan sampai pada penyelesaian skripsi ini, penulis telah banyak memperoleh bimbingan, bantuandan saran yang sangat bermanfaat, baik secara moril maupun materiil dari berbagai pihak. Untuk itu pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih kepada :

1. Prof. Dr. Ir. Abraham Lommi, MSEE, selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. Sidik Noertjahjono, MT, selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang.
3. Bapak Ir. F Yudi Limpraptono, MT, selaku Ketua Jurusan Teknik Elektro Institut Teknologi Nasional Malang.
4. Bapak Ir. F Yudi Limpraptono, MT, selaku dosen pembimbing, yang telah banyak meluangkan waktu, fikiran, arahan dan tuntunan yang progesif dalam penyusunan skripsi ini.
5. Bapak Ismail Yusuf Nahkoda, MT, selaku Sekretaris Jurusan Teknik Elektro Institut Teknologi Nasional Malang.
6. Seluruh rekan-rekan yang telah membantu penulis dalam penyelesaian skripsi ini.

Akhirnya penulis menyadari bahwa skripsi ini masih ada kekurangan Kekurangan, namun dmikian penulis mengharapkan pandangan-pandangan dan saran-saran untuk menyempurnakan dalam pembuatan alat serta penulisan skripsi ini, baik dari pihak mahasiswa Institut Teknologi Nasional Malang, maupun dari pihak lain yang menggunakan skripsi ini sebagai bahan acuan dan referensi.

Semoga ALLAH SWT, selalu melimpahkan Taufik dan Hidayah-Nya kepada semua pihak yang telah membantu dalam penyusunan skripsi ini.

Malang, Maret 2010

Penulis

ABSTRAK

Penulis : DIDI AFFANDI
NIM : 98.17.006

PERANCANGAN DAN PEMBUATAN GENERATOR SINYAL UNTUK SISTEM PENGUJI RANGKAIAN ELEKTRONIK BERBASIS MIKROKONTROLLER AT89C51

Generator sinyal merupakan salah satu kunci peralatan penting karena fungsinya sebagai pembangkit sinyal. DDS (direct digital synthesis) dapat digunakan untuk membuat function generator dengan harga yang murah dan ukuran yang kecil. Penyusunan skripsi ini bertujuan untuk mendapatkan perancangan dan pembuatan generator sinyal untuk sistem penguji rangkaian elektronik berbasis mikrokontroller AT89C51.

Penyusunan skripsi dilakukan dengan merancang dan membuat generator sinyal untuk sistem penguji rangkaian elektronik berbasis mikrokontroller AT89C51. Perancangan dan pembuatan meliputi hardware dan software. Hardware meliputi komponen pembangkit sinyal berbasis DDS (Direct Digital Syntesis), keypad, DAC R2R tipe ladder. Bahasa pemrograman yang digunakan di computer adalah bahasa Delphi. Hasil pembuatan alat diuji coba untuk mengetahui kinerja system yang telah dirancang. Uji coba dilakukan dengan melakukan pengujian terhadap rangkaian elektronik.

Hasil pengujian menunjukkan bahwa sistem dapat bekerja dengan baik. Rangkaian DAC sudah akurat, tetapi hasil pengukuran menunjukkan tingkat error rata-rata 0,08%. Dari hasil pengujian alat, bentuk gelombang sinusoidal masih belum sempurna karena tidak melalui proses filter. Karena keterbatasan mikrokontroller AT89C51, sistem ini hanya akurat untuk frekuensi tertentu. Generator sinyal untuk sistem penguji rangkaian elektronik berbasis mikrokontroller AT89C51 telah berfungsi tetapi masih memerlukan penyempurnaan lebih lanjut.

DAFTAR ISI

DAFTAR GAMBAR

BAB I

PENDAHULUAN

1.1. Latar Belakang

Pemakaian teknologi elektronika sangat luas dalam kehidupan manusia sehari-hari, baik di skala rumah tangga maupun di dunia industri. Salah satu teknologi elektronika yang berkembang pesat saat ini adalah teknologi sinyal audio. Sinyal audio banyak digunakan di bidang telekomunikasi seperti telepon, pertelevisian, radio dan lain-lain.

Generator sinyal merupakan salah satu kunci peralatan penting karena fungsinya sebagai pembangkit sinyal. Di dunia elektronik *function generator* sudah tidak asing lagi. Akan tetapi karena harganya sangat mahal. DDS *(direct digital synthesis)* dapat digunakan untuk membuat *function generator* dengan harga yang murah dan ukuran yang kecil.

System DDS (*Direct Digital Synthesizer*) merupakan sebuah teknik pemrosesan data digital untuk membangkitkan frekuensi secara langsung. Inti dari system ini adalah arsitektur akumulator dengan resolusi mencapai mili Hertz dan frekuensi sinyal yang dihasilkan dapat diatur tergantung dari sinyal frekuensi referensi dan metode perancangan. Keluaran sistem DDS yang diproses oleh Mikrokontroler berupa sinyal digital kemudian menjadi masukan untuk DAC *(D/A converter)* untuk menghasilkan sinyal sinusoidal, kotak dan segi tiga yang sempurna.

2. Rumusan Masalah

Rumusan masalah dalam penyusunan skripsi ini adalah bagaimana perancangan dan pembuatan generator sinyal untuk sistem penguji rangkaian elektronik berbasis mikrokontroller AT89C51.

3. Tujuan

Tujuan penyusunan skripsi ini adalah merancang dan membuat generator sinyal untuk sistem penguji rangkaian elektronik berbasis mikrokontroller AT89C51.

4. Batasan Masalah

Agar penyusunan skripsi ini lebih terarah maka perlu dilakukan pembatasan masalah. Adapun batasan masalahnya dalam hal ini adalah :

1. Mikrokontroller yang digunakan adalah mikrokontroller AT89C51.
2. Generator sinyal yang dirancang adalah generator sinyal berbasis Direct Digital Synthesis (DDS).
3. Input data yang digunakan dalam pengujian sistem berbentuk sinyal audio.
4. Tidak membahas rangkaian elektronika yang diuji.

1.5. Metode Penelitian

Metode penyusunan skripsi ini adalah :

1. Studi literatur untuk mendapatkan teori-teori yang sesuai.

   Yaitu aktivitas mengumpulkan referensi atau teori-teori yang sesuai dari buku-buku atau sumber internet.

2. Perancangan generator sinyal.

   Dalam hal ini dibahas tentang langkah-langkah perancangan sistem, blok diagram dan spesifikasi komponen sistem.

3. Pembuatan alat.

   Yaitu aktivitas pembuatan sistem yang telah dirancang.

4. Pengujian sistem.

   Yaitu aktivitas pengujian sistem untuk mendapatkan kinerja sistem yang telah dibuat.

5. Kesimpulan.

   Yaitu ringkasan seluruh hasil perancangan, pembuatan dan pengujian sistem.

1.6. Sistematika Penulisan

Sistematika penulisan skripsi ini adalah :

BAB I PENDAHULUAN

Membahas tentang latar belakang, rumusan masalah, tujuan, batasan masalah dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Membahas teori-teori yang bersesuaian dengan topik skripsi ini.

BAB III PERANCANGAN, PEMBUATAN DAN PENGUJIAN ALAT

Membahas proses perencanaan, pembuatan dan pengujian alat.

BAB IV PENGUJIAN SISTEM

Membahas kinerja alat yang telah dirancang dan dibuat serta dianalisa berdasarkan hasil yang diperoleh.

BAB V KESIMPULAN

Yaitu ringkasan hasil perancangan, pembuatan dan pengujian alat.

DAFTAR TABEL

BAB II

TINJAUAN PUSTAKA

2.1. Pendahuluan

Bab ini menjelaskan tentang pembangkitan sinyal, Direct Digital Synthesis, Mikrokontroller AT89C51 dan DAC R2R ladder.

2.2. Pembangkitan Sinyal

Sinyal merupakan bentuk input yang banyak dibutuhkan dalam dunia elektronika misalnya media telekomunikasi. Hal ini menyebabkan perkembangan peralatan yang berfungsi sebagai pembangkit sinyal. Pembangkitan sinyal merupakan salah satu aplikasi teknik elektronika yang banyak digunakan.

Pembangkit sinyal (*signal generator*) adalah instrumen elektronik yang menghasilkan gelombang sinus yang kontinu. Instrumen ini juga sering disebut sebagai *test signal generator, tone generator* untuk peralatan audio, pembangkit gelombang atau pembangkit frekuensi.

Gelombang yang dibangkitkan dapat berupa gelombang analog maupun gelombang digital. Instrumen ini umumnya digunakan untuk perencanaan, pengujian, *troubleshooting*, dan perbaikan komponen elektronik. Sebuah oscillator elektronik digunakan untuk membentuk gelombang yang dibangkitkan. Gelombang yang umum digunakan adalah gelombang sinus. Jenis-jenis gelombang lain yang juga digunakan adalah segitiga atau kotak.

2.3. Direct Digital Synthesis

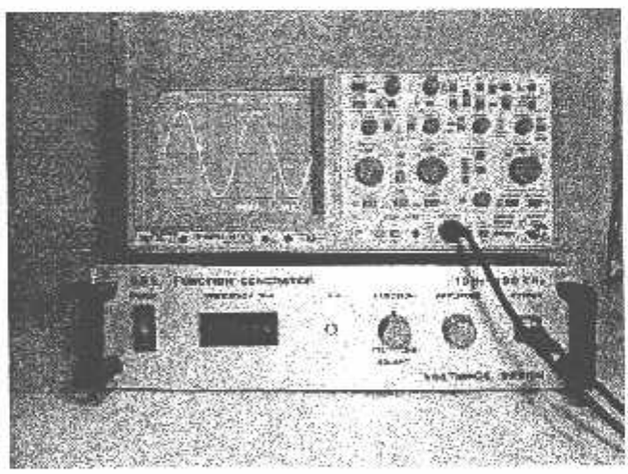
Direct Digital Synthesis adalah sinyal generator tipe function yang banyak digunakan saat ini. Generator sinyal ini merupakan suatu metode untuk menghasilkan gelombang berbentuk analog (umumnya gelombang sinus) dengan cara membangkitkan sebuah sinyal berbasis perubahan waktu dalam bentuk digital dan kemudian diproses dengan sebuah konverter digital ke analog (Digital Analog Converter = DAC). DDS juga sering disebut sebagai oscilator yang dikontrol secara numerik atau disingkat NCO (*numerically controlled oscilator*)

DDS pada awal perkembangannya hanya terbatas pada pengaturan amplitudo dan frekuensi sebuah gelombang sinus. DDS memberikan peluang untuk dikembangkan dalam aplikasi yang lebih luas. Saat ini, generator sinyal berbasis DDS memungkinkan untuk didesain dalam area frekuensi yang lebih luas, level output yang terkalibrasi, variasi bentuk gelombang, mode modulasi, koneksi dengan komputer dan lain-lain.

System DDS (*Direct Digital Synthesizer*) merupakan sebuah teknik pemrosesan data digital untuk membangkitkan frekuensi secara langsung. Inti dari system ini adalah arsitektur akumulator dengan resolusi mencapai mili Hertz dan frekuensi sinyal yang dihasilkan dapat diatur tergantung sinyal frekuensi referensi dan metode perancangan. Keluaran sistem DDS yang diproses oleh mikrokontroler berupa sinyal digital kemudian menjadi masukan untuk DAC *(D/A converter)* untuk menghasilkan sinyal sinusoidal, kotak dan segi tiga yang sempurna.

Semua parameter kontrol sistem DDS berada dalam bentuk besaran digital. Sistem DDS pada dasarnya terdiri atas akumulator phasa, LUT (*Look Up Table*), dan osilator sebagai pembangkit frekuensi referensi *(clock)*. Sedangkan DAC *(digital to Analog Converter)* merupakan komponen-komponen penunjang sistem DDS.

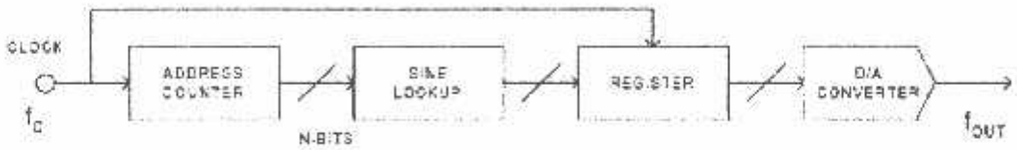
Kelebihan penggunaan sistem DDS adalah Karakteristik sistem DDS itu sendiri, dimana keutamaan dari sistem ini adalah memiliki *settling time* / kecepatan yang cepat dan memiliki resolusi frekuensi yang halus terhadap frekuensi keluaran, operasi atas suatu spektrum frekuensi yang lebar dan dengan kemajuan dalam disain teknologi proses, serta sangat ringkas dan membutuhkan sedikit daya. Sehingga sangat memungkinkan sistem DDS bisa lebih dikembangkan untuk disain alat yang berkaitan dengan aplikasi-aplikasi *frequency hopping* serta sistem-sistem yang berkaiatan dengan peralatan pemancar radio,TV, peralatan test, dan lain-lain.



Gambar 2.1 : Contoh tampilan DDS [11]

2.4. Prinsip Dasar DDS

Secara sederhana, sebuah DDS dapat digambarkan sebagai sebuah sistem yang terdiri dari sebuah penunjuk waktu referensi yang presisi (*precision reference clock*), sebuah penghitung alamat (*address counter*), sebuah PROM (*programmable read only memory*) dan dan sebuah konverter digital ke analog (*D/A converter*). Sistem ini dapat dilihat pada gambar berikut ini :



Gambar 2.2 : Skema DDS sederhana [1]

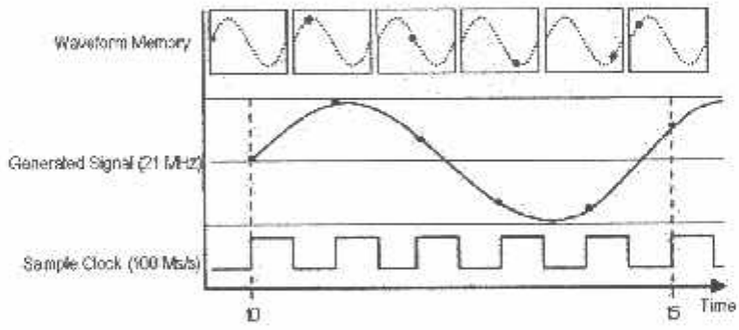
Dari gambar di atas, informasi amplitudo digital yang berhubungan dengan sebuah gelombang sinus disimpan dalam PROM. PROM dalam hal ini berfungsi sebagai tabel penyimpan gelombang sinus tersebut. *Address counter* mengakses setiap lokasi memori PROM dan isinya (dalam hal ini gelombang sinus) dan ditampilkan ke konverter digital ke analog. Konverter ini selanjutnya membangkitkan sebuah gelombang sinus sebagai respon terhadap input digital dari PROM. Frekuensi keluaran dari sistem DDS di atas tergantung pada :

1. Frekuensi *reference clock*.
2. Ukuran gelombang sinus yang diprogram ke dalam PROM.

Frekuensi output hanya dapat dirubah dengan merubah frekuensi *reference clock* atau dengan memprogram ulang PROM.

2.5. Penggunaan Memori DDS

Generator fungsi (*function generator*) menggunakan DDS untuk membangkitkan sinyal periodik pada frekuensi tertentu dengan memilih sampel dari memori daripada membangkitkan semua sampel dari sebuah gelombang. Kebalikannya, pembangkit gelombang arbitari (*arbitary waveform generator* = AWG) membangkitkan setiap sampel sebuah gelobang yang tersimpan ke dalam memori. Sementara AWG memperbolehkan penggunanya untuk mendefinisikan secara khusus sebuah gelombang yang sedang dibangkitkan, terdapat batasan dalam keakuratan frekuensi yang dapat dicapai, khususnya pada frekuensi tinggi. Ilustrasi bagaimana sebuah generator fungsi dapat membangkitkan sebuah gelombang sinus 21 MHz dapat dilihat pada gambar 2.3, meskipun frekuensinya bukan merupakan jumlah langsung dari kecepatan sampelnya.



Gambar 2.3 : Pembangkitan gelombang sinus 21 MHz dengan DDS [6]

Dari gambar di atas terlihat bahwa frekuensi gelombang sinusnya bukan merupakan pembagi dari kecepatan sampelnya. Akibatnya, membangkitkan gelombang tersebut akan sulit dengan menggunakan AWG pada 1,00 MS/s. Generator fungsi menggunakan DDS untuk menyimpan 16.384 sampel

gelombang dalam memori. Untuk setiap siklus waktu, sampel yang sesuai dipilih dari tabel memori dan kemudian dibangkitkan. Hasilnya, sebuah sinyal dapat dibangkitkan pada frekuensi yang akurat bersamaan dengan pengirimannya ke konverter digital-analog (DAC) dengan frekuensi konstan sebesar 100 MHz.

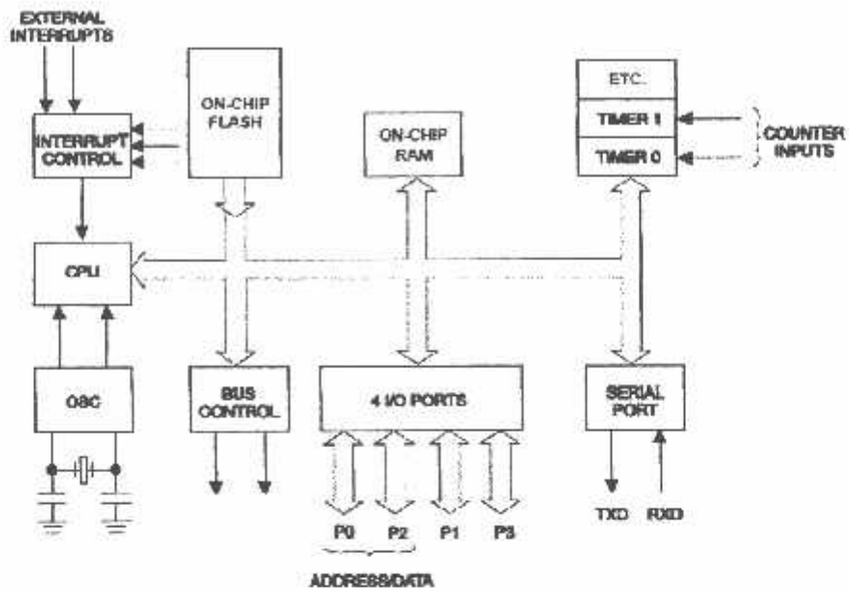
2.6. Mikrokontroller AT89C51

Mikrokontroler AT89C51 ialah mikrokomputer CMOS 8 bit dengan 4KB *Flash Programmable dan Erasable Read Only Memory* (PEROM). Mikrokontroler berteknologi memori *non volatile* kerapatan tinggi dari Atmel ini kompatibel dengan mikrokontroler standar industri MCS-51 (seperti mikrokontroler 8031 yang terkenal dan banyak digunakan beberapa waktu lalu) baik pin kaki IC maupun set instruksinya serta harganya yang cukup murah.

Gambar 2.4: Mikrokontroller AT89C51 [4]

AT89C51 mempunyai memori yang terdiri dari RAM internal 128 byte dengan alamat 00H-7FH dapat diakses menggunakan RAM address register. RAM ini terdiri dari *Register Banks* dengan 8 buah register (R0-R7).Memori lain yaitu 21 buah *Special Function Register* dimulai dari alamat 80H-FFH. RAM ini beda lokasi dengan Flash PEROM dengan alamat 000H -7FFH.

2.6.1. Arsitektur AT89C51



Gambar 2.5: Arsitektur mikrokontroller AT89C51 [10]

Untuk penghematan power dalam CMOS, mikrokontroller Flash Atmel memiliki 2 software untuk mengatur power mode yaitu :

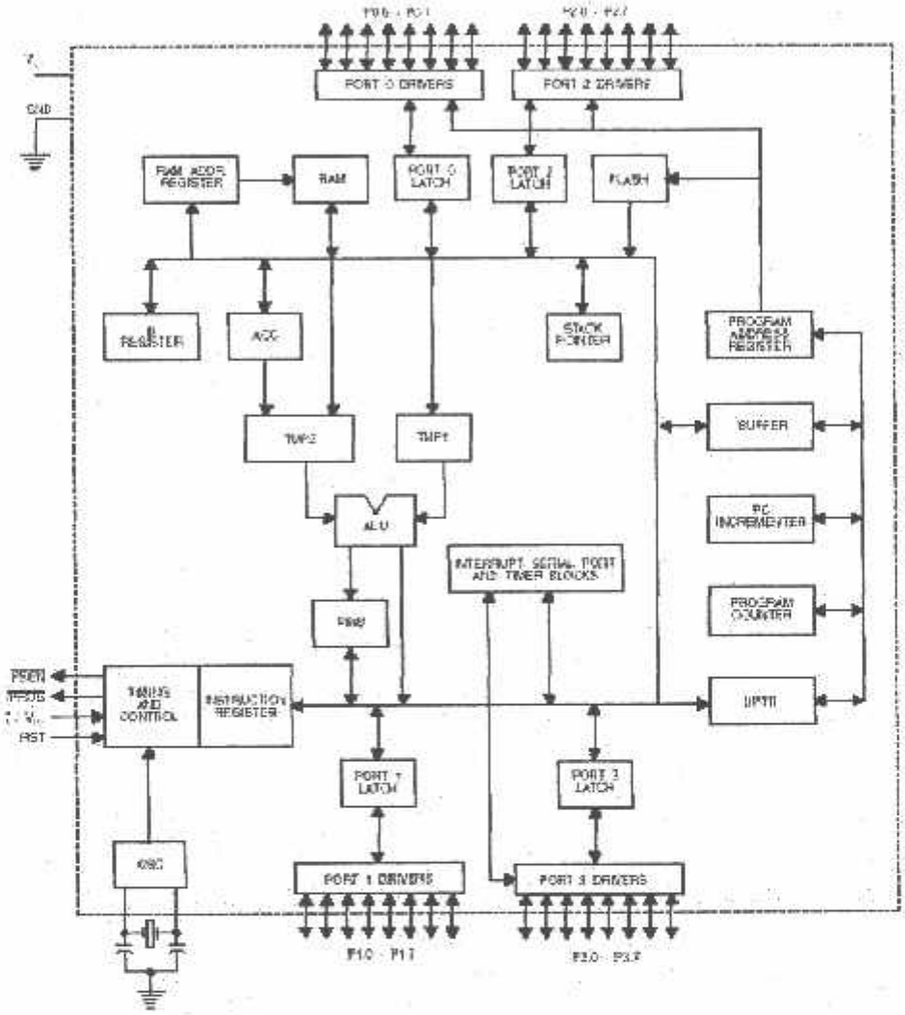
1. Idle mode.

   CPU dimatikan sementara RAM dan peripheral chip lainnya tetap beroperasi. Dalam mode ini, arus listrik berkurang sekitar 15% dibanding jika CPU dalam kondisi aktif.

2. Power down mode.

   Semua chip tidak diaktifkan sementara chip RAM tetap menahan data. Dalam mode ini, peralatan memerlukan kurang dari 15 μA dan bahkan dapat serendah 0,6 μA.

Sebagai tambahan, peralatan ini didesain menggunakan logika statis yang tidak memerlukan clocking terus-menerus. Frekuensi clock dapat dipelankan atau bahkan dihentikan sementara menunggu untuk aktivitas internal.



Gambar 2.6: Blok diagram AT89C51 [10]

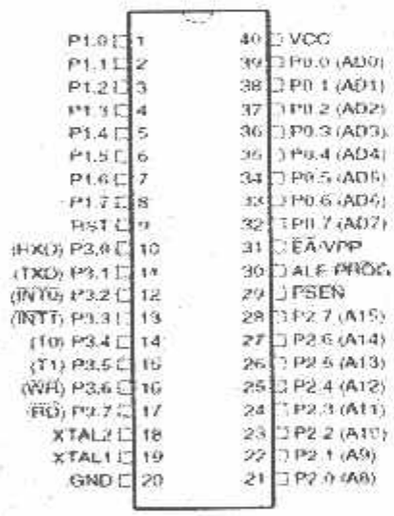
Mikrokontroller AT89C51 memiliki ciri standar sebagai berikut :

1. Flash 4K bytes.
2. RAM 128 bytes

3. 32 I/O lines
4. Dua buah timer/counter 16-bit
5. Interupsi 2 level sebanyak 5 vektor.
6. Full duplex serial port.
7. On-chip ascilator dan clock circuitry.

Sebagai tambahan, mikrokontroller ini didesain dengan logika statis untuk operasi sampai frekuensi nol dan mendukung dua perangkat lunak yang dipilih dalam kondisi *power saving mode*. Idle Mode dapat menghentikan CPU sementara RAM, timer/counter, serial port dan sistem interupsi tetap berfungsi.

2.6.2. Konfigurasi Pin



Gambar 2.7: Konfigurasi pin AT89C51 [10]

Deskripsi tiap pin AT89C51 :

1. VCC.

   Suplai tegangan

2. GND.

   Ground.

3. Port 0.

   Port 0 adalah sebuah open-drain bi-directional I/O port 8 bit. Setiap pin dapat menerima 8 input TTL. Ketika nomor 1 menulis data ke pin port 0, pin-pin tersebut dapat digunakan sebagai input impendansi tinggi. Port 0 juga dapat dikonfigurasikan menjadi multiplexed alamat atau data orde rendah saat mengakses program eksternal dan memori data. Pada mode ini, port 0 memiliki internal pull up. Port 0 juga menerima kode bytes selama pemrograman Flash, dan mengeluarkan output selama verifikasi program. Eksternal pull up diperlukan selama verifikasi ini.

4. Port 1.

   Port 1 adalah scbuah port bi-directional I/O 8 bit dengan internal pull up. Port 1 mengeluarkan atau menerima 4 input TTL. Port 1 juga menerima alamat berorde bytes rendah selama pemrograman Flash dan verifikasi.

5. Port 2.

   Port 2 adalah sebuah port bi-directional I/O 8 bit dengan internal pull up. Port 2 mengeluarkan alamat berorde byte tinggi dari memori program eksternal dan selama mengakses data memori menggunakan alamat 16 bit (MOVX @DPTR). Selama mengakses memori data eksternal yang menggunakan alamat 8 bit (MOVX@RI), port 2 mengeluarkan content dari P2 Special Function Register. Port 2 juga meerima alamat berorde bit tinggi dan beberapa sinyal kontrol selama pemrograman Flash dan verifikasi.

6. Port 3.

Port 2 adalah sebuah port bi-directional I/O 8 bit dengan internal pull up. Port 3 juga menyediakan fungsi-fungsi khusus berikut :

Tabel 2.1 : Fungsi khusus port 3 mikrokontroller AT89C51 [10]

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{\text{INT0}}$ (external interrupt 0)
P3.3	$\overline{\text{INT1}}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{\text{WR}}$ (external data memory write strobe)
P3.7	$\overline{\text{RD}}$ (external data memory read strobe)

Port 3 juga menerima beberapa sinyal kontrol selama pemrograman Flash dan verifikasi.

7. RST.

Input reset.

8. $\overline{\text{ALE}}$/PROG

Address Latch Enable megeluarkan pulsa untuk mengunci byte rendah dari alamat selama mengakses memori eksternal. Pin ini juga merupakan input program pulsa selama pemrograman Flash. Pada operasi normal, ALE mengeluarkan data pada laju tetap sebesar 1/6 frekuensi osilasi dan dapat digunakan untuk timing eksternal atau tujuan clocking.

9. $\overline{\text{PSEN}}$

Program Store Enable adalah unit pembaca untuk program memori eksternal. Ketika AT89C51 menerima kode dari program eksternal, PSEN diaktifkan dua kali tiap siklusnya, kecuali aktivasi ini dimatikan.

10. $\overline{EA}$/VPP.

External Access Enable. EA harus dihubungkan pada GND agar perangkat ini dapat menerima kode dari memori program eksternal yang lokasinya mulai 0000H sampai FFFFH. EA harus dihubungkan dengan Vcc untuk eksekusi program internal. Pin ini juga menerima tegangan 12 Volt (Vpp) selama pemrograman Flash untuk komponen yang membutuhkan tegangan tersebut.

11. XTAL1.

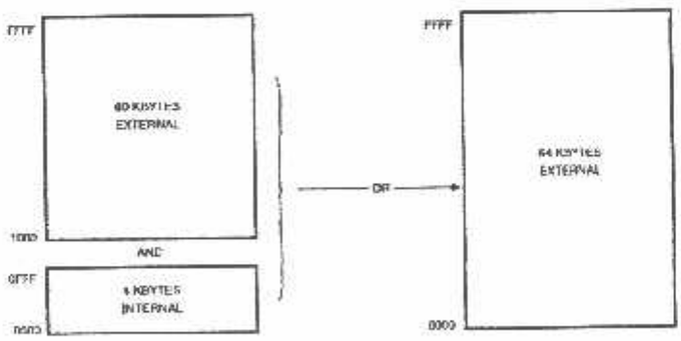
Berfungsi untuk input pada inverting oscillator amplifier dan internal clock operating circuit.

12. XTAL2

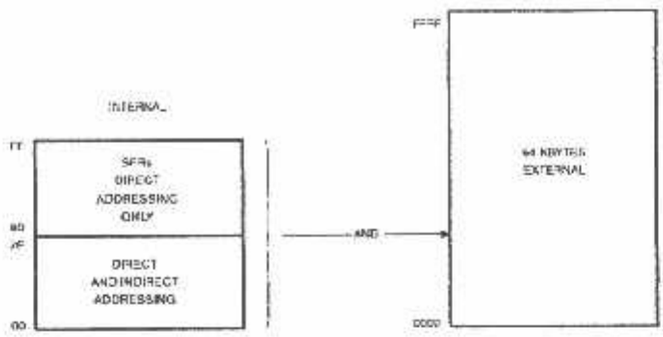
Berfungsi untuk output pada inverting oscillator amplifier.

2.6.3. Program Memori

Mikrokontroller AT89C51 memiliki ruang alamat yang terpisah untuk program memori dan data memori. Program memori dapat berukuran lebih dari 64 kilo bytes.



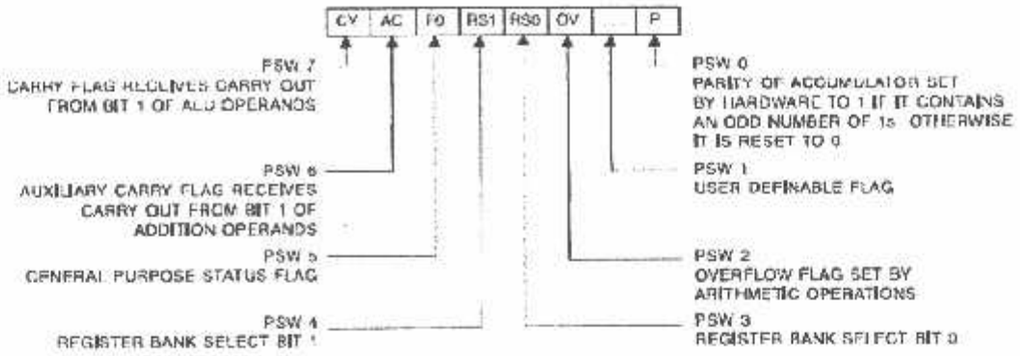
Gambar 2.8 : Program memori AT89C51 [10]



Gambar 2.9: Data memori AT89C51 [10]

2.6.4. PSW (Program Status Word) Register

PSW berisi bit status yang menunjukkan kondisi arus dalam CPU. PSW terdiri dari Carry bit, auxiliary carry, 2 bank register select bits, overflow flag, parity bit dan 2 status flag pengguna. Carry bit berfungsi sebagai operasi aritmetik dan sebagai accumulator untuk sejumlah Boolean. Bit RS0 dan RS1 memilih satu dari 4 bank register. Sejumlah instruksi mengacu pada lokasi RAM sebagai R0 melalui R7. Status dari RS0 dan RS1 bit pada waktu eksekusi menunjukkan bank register mana yang dipilih. Parity bit merefleksikan sejumlah 1s pada accumulator. P=1 jika accumulator berisi sebuah angka ganjil dari 1s dan P=0 jika berisi sebuah angka genap dari 1s. Sehingga, jumlah dari 1s di accumulator ditambah P selalu genap.



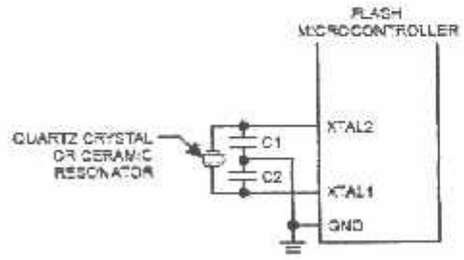
Gambar 2.10 : PSW register [10]

Tabel 2.2 : Nilai RS0 dan RS1 [10]

RS1	RS0	Register Bank	Address
0	0	0	00H-07H
0	1	1	08H-0FH
1	0	2	10H-17H
1	1	3	18H-1FH

2.6.5. Oscillator

AT89C51 memiliki sebuah on-chip oscillator yang dapat digunakan sebagai sumber clock untuk CPU. Chip ini dihubungkan dengan resonator kristal atau keramik antara pin XTAL1 dan XTAL2 dan dihubungkan dengan kapasitor ke ground. Pembangkit clock internal mendefinisikan tahapan kondisi saat mikrokontroller melakukan siklusnya.



Gambar 2.11 : Koneksi oscillator [10]

2.6.6. Interupsi

AT89C51 memiliki 5 sumber interupsi yaitu 2 interupsi eksternal, 2 interupsi timer dan interupsi serial port.

1. Interrupt Enables.

   Setiap sumber interupsi secara individual dapat dijalankan atau dimatikan dengan mengatur bit Interrupt Enable (IE) di SFR. Register ini juga terdiri dari sebuah global disable bit, yang dapat mematikan semua interupsi secara bersamaan.

Tabel 2.3 : Interrupt Enable Register [10]

(MSB)							(LSB)
EA	—	—	ES	ET1	EX1	ET0	EX0

Enable bit = 1 enables the interrupt.

Enable bit = 0 disables it.

Symbol	Position	Function
EA	IE.7	Disables all interrupts. If EA = 0, no interrupt will be acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
—	IE.6	reserved.*
—	IE.5	reserved.*
ES	IE.4	Serial Port interrupt enable bit.
ET1	IE.3	Timer 1 Overflow interrupt enable bit.
EX1	IE.2	External interrupt 1 enable bit.
ET0	IE.1	Timer 0 Overflow interrupt enable bit.
EX0	IE.0	External interrupt 0 enable bit.

*These reserved bits are used in other Atmel microcontrollers.

2. Interrupt Priorities.

   Setiap sumber interupsi juga dapat diprogram secara individual untuk satu atau 2 level prioritas dengan mengatur bit Interrupt Priority (IP). Interupsi low-priority dapat diinterupsi dengan interupsi high-priority tetapi tidak dengan low-priority yang lain.

Tabel 2.4 : Interrupt Priority Register [10]

(MSB)							(LSB)
—	—	—	PS	PT1	PX1	PT0	PX0

Priority bit = 1 assigns high priority.

Priority bit = 0 assigns low priority.

Symbol	Position	Function
—	IP.7	reserved.*
—	IP.6	reserved.*
—	IP.5	reserved.*
PS	IP.4	Serial Port Interrupt priority bit.
PT1	IP.3	Timer 1 interrupt priority bit.
PX1	IP.2	External Interrupt 1 priority bit.
PT0	IP.1	Timer 0 Interrupt priority bit.
PX0	IP.0	External Interrupt 0 priority bit.

*These reserved bits are used in other Atmel microcontrollers.

2.5.7. Timer/Counter

1. TCON : Timer/Counter Control Register (Bit Addressable).

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TF1 : Timer 1 overflow flag

TR1 : Timer 1 run control bit

TF0 : Timer 0 overflow flag

TR0 : Timer 0 control bit

IE1 : Internal Interrupt edge flag

IT1 : Interupsi 1 tipe control bit

IE0 : External Interrupt 0 edge flag

IT0 : Interupsi 0 tipe control bit

2. TMOD : Timer/Counter Mode Control Register (Not Bit Addressable)

Timer 1				Timer 0			
GATE	C/T	M1	M0	GATE	C/T	M1	M0

GATE : Jika TRx (di TCON) diatur dan GATE = 1, timer/counterx beroperasi hanya jika pin INTx kondisi high.

C/$\overline{T}$: Timer atau counter selector

M1 : Mode selector bit

M0 : Mode selector bit

Pengaturannya adalah sebagai berikut :

Tabel 2.5 : Pengaturan TMOD [10]

M1	M0	Operating Mode	
0	0	0	13-bit Timer
0	1	1	16-bit Timer/Counter
1	0	2	8-bit Auto-Reload Timer/Counter
1	1	3	Split Timer Mode: (Timer 0) TL0 is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits. TH0 is an 8-bit Timer and is controlled by Timer 1 control bits.
1	1	3	(Timer 1) Timer/Counter 1 stopped.

3. Timer Set uap

Tabel-tabel berikut menunjukkan nilai TMOD yang dapat digunakan untuk mengatur timer 0 dalam mode berbeda. Diasumsikan bahwa hanya satu timer yang digunakan pada satu waktu. Jika timer 0 dan 1 harus bekerja bersama-sama (simultan) maka nilai TMOD untuk timer 0 harus ORed dengan nilai yang ditunjukkan oleh timer 1. Misalnya, jika timer 1 bekerja di mode GATE (kontrol eksternal) dan timer 1 harus dijalankan pada mode 2 Counter maka nilai yang harus dimasukkan ke dalam TMOD adalah 69H.

Tabel 2.6 : Timer/Counter 0 digunakan sebagai Timer [10]

MODE	TIMER 0 FUNCTION	TMOD	
		INTERNAL CONTROL(1)	EXTERNAL CONTROL(2)
0	13-bit Timer	00H	08H
1	16-bit Timer	01H	09H
2	8-bit Auto-Reload	02H	0AH
3	two 8-bit Timers	03H	0BH

Tabel 2.7 : Timer/Counter 0 digunakan sebagai Counter [10]

MODE	TIMER 0 FUNCTION	TMOD	
		INTERNAL CONTROL(1)	EXTERNAL CONTROL(2)
0	13-bit Timer	04H	0CH
1	16-bit Timer	05H	0DH
2	8-bit Auto-Reload	06H	0EH
3	one 8-bit Counter	07H	0FH

Tabel 2.8 : Timer/Counter 1 digunakan sebagai Timer [10]

MODE	TIMER 1 FUNCTION	TMOD	
		INTERNAL CONTROL(1)	EXTERNAL CONTROL(2)
0	13-bit Timer	00H	80H
1	16-bit Timer	10H	90H
2	8-bit Auto-Reload	20H	A0H
3	does not run	30H	B0H

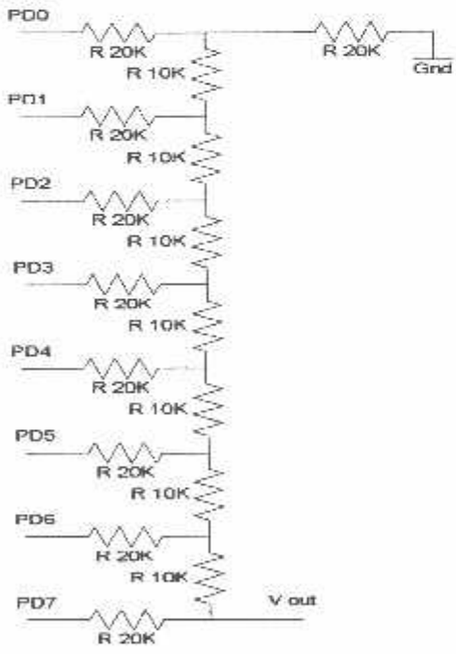
Tabel 2.9 : Timer/Counter 1 digunakan sebagai Counter [10]

MODE	COUNTER 1 FUNCTION	TMOD	
		INTERNAL CONTROL(1)	EXTERNAL CONTROL(2)
0	13-bit Timer	40H	C0H
1	16-bit Timer	50H	D0H
2	8-bit Auto-Reload	60H	E0H
3	not available	—	—

2.7. Digital Analog Converter

DAC (Digital to Analog) adalah piranti pengubah sinyal digital menjadi sinyal analog. DAC memiliki arsitektur yang bervariasi, antara lain DAC R2R ladder dan DAC PWM (1 bit). DAC R2R ladder merupakan bentuk yang paling sederhana dari IC DAC yang banyak beredar di pasaran. DAC ini terdiri atas resistor yang dipasang berbentuk tangga. Nilai resistor yang dipergunakan adalah R dan 2R.

Dengan memberikan nilai digital (biner) pada masukannya, akan diperoleh nilai analog yangs sesuai pada keluarannya. Kelinieran dari DAC ini bergantung pada ketepatan nilai resistor yang digunakan. DAC ini biasanya memiliki offset (nilai digital 0 tidak memberikan nilai analog 0 V) serta nilai fullscale (tegangan analog maksimum).



Gambar 2.12 : Rangkaian R-2R [2]

BAB III

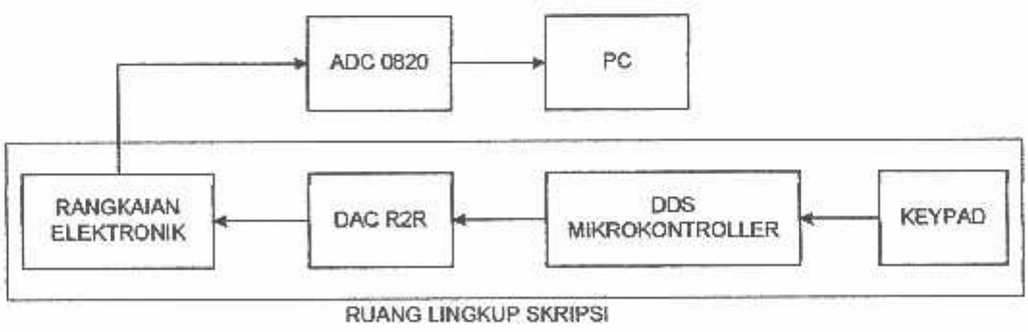
PERENCANAAN DAN PEMBUATAN ALAT

3.1. Pendahuluan

Pada Bab II telah dijelaskan berbagai teori penunjang yang digunakan dalam perancangan sistem dalam skripsi ini. Selanjutnya tahap-tahap perencanaan seluruh sistem. Sebagai acuan awal maka digunakan spesifikasi dasar perencanaan system. Adapun sistem yang direncanakan didasarkan pada beberapa spesifikasi sebagai berikut :

a. Menggunakan mikrokontroller AT89C51 yang difungsikan sebagai pembangkit sinyal berbasis DDS.
b. Konversi data atau sinyal digital menjadi data analog menggunakan DAC jenis ladder (R2R).
c. Perintah kepada mikrokontroller dilakukan dengan keypad.

3.2. Blok Diagram Sistem



Gambar 3.1 : Blok diagram sistem

Fungsi dari masing-masing subsistem pada blok diagram di atas adalah sebagai berikut :

1. Keypad.

   Untuk memberikan perintah pada mikrokontroller, terdiri dari tombol-tombol :

   - *Start* berfungsi untuk memulai proses membangkitkan signal setelah pengesetan frekuensi dilakukan
   - *Reset* berfungsi untuk menghentikan proses pembangkitan frekuensi
   - *Up* berfungsi untuk menambahkan nilai frekuensi pada waktu *setting*
   - *Down* berfungsi untuk mengurangi nilai frekuensi pada waktu *setting*
   - *Mode1* berfungsi untuk menentukan kenaikan step nilai frekuensi
   - *Mode2* berfungsi untuk memilih / menentukan *output* gelombang

2. PC.

   Digunakan sebagai unit yang mengontrol atau memberikan perintah pada mikrokontroller.

3. Mikrokontroller AT89C51.

   Mikrokontroller dalam system ini memiliki 2 fungsi yaitu sebagai DDS atau pembangkit sinyal dan pengontrol gerakan motor stepper.

4. DAC.

   Komponen ini berfungsi untuk merubah sinyal digital ke dalam bentuk analog. Jenis DAC yang digunakan dalam hal ini adalah jenis ladder.

3.2. Perencanaan Perangkat Keras

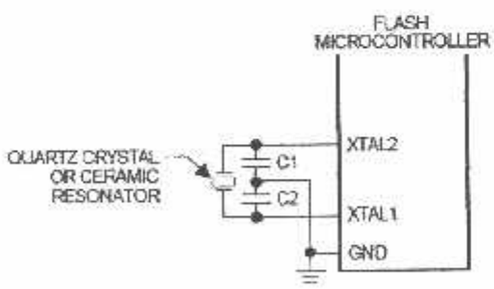
3.2.1. Perencanaan Mikrokontroller AT89C51

3.2.1.1. Sistem Minimum Mikrokontroller AT89C51

Pada rangkaian ini komponen utamanya adalah unit Mikrokontroler AT89C51. Komponen ini merupakan sebuah *chip* tunggal sebagai pengolah data dan pengontrolan alat yang tidak memerlukan memori luar. Kristal yang digunakan untuk mengoperasikan mikrokontroler adalah 12 MHz. Penggunaan kristal 12 MHz menyebabkan detak dalam pada mikrokontroler menjadi 12 Mhz / 12 = 1 MHz, yang artinya setiap periode detak waktunya 1 mikrodetik, sehingga memudahkan untuk mengubah-ubah data pada penggunaan pewaktu karena periode detak pewaktuannya tidak ganjil.

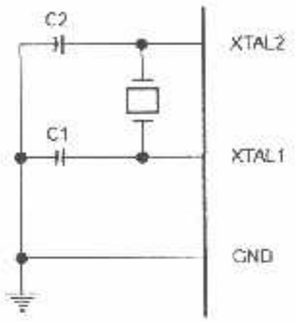
3.2.1.2. Koneksi Dengan CPU

Semua mikrokontroller Atmel Flash memiliki sebuah oscillator on-chip, yang dapat digunakan sebagai sumber clock untuk CPU. Untuk menggunakan chip ini, resonator kristal atau keramik dihubungkan yaitu antara pin-pin XTAL1 dan XTAL2 di mikrokontroller. Kapasitor selanjutnya dihubungkan ke ground seperti pada gambar berikut :



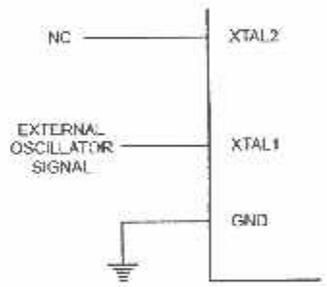
Gambar 3.2 : Penggunaan oscillator on-chip [10]

Cara mengendalikan clock dengan oscillator eksternal dapat dilihat pada gambar berikut :



Gambar 3.3 : Koneksi oscillator [10]

Pembangkit clock internal mendefinisikan kondisi tahapan mikrokontroller dalam melakukan satu siklus.

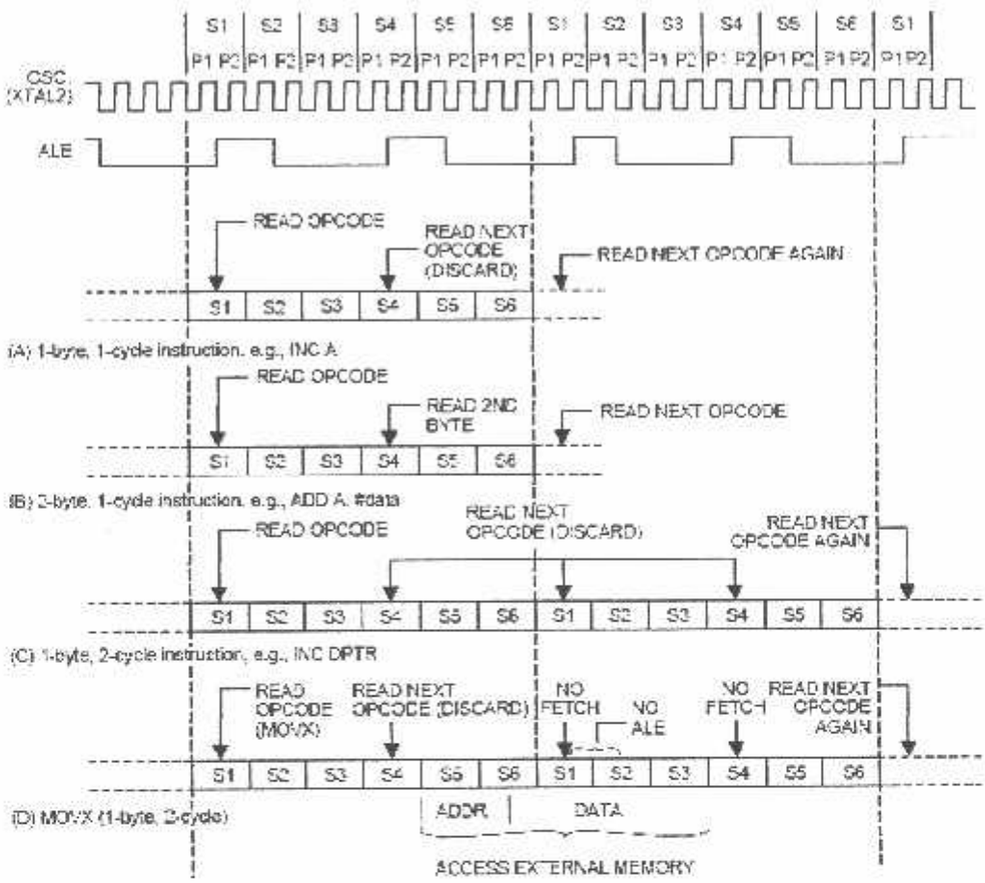


Gambar 3.4 : Konfigurasi clock drive eksternal [10]

3.2.1.3. Timing Diagram

Setiap siklus pada mikrokontroller AT89C51 terdiri dari tahapan 6 kondisi yang diberi nomor S1 sampai S6. Setiap kondisi waktu ditahan untuk dua periode oscillator. Maka dari itu, satu siklus menahan 12 periode oscillator atau 1 μs jika

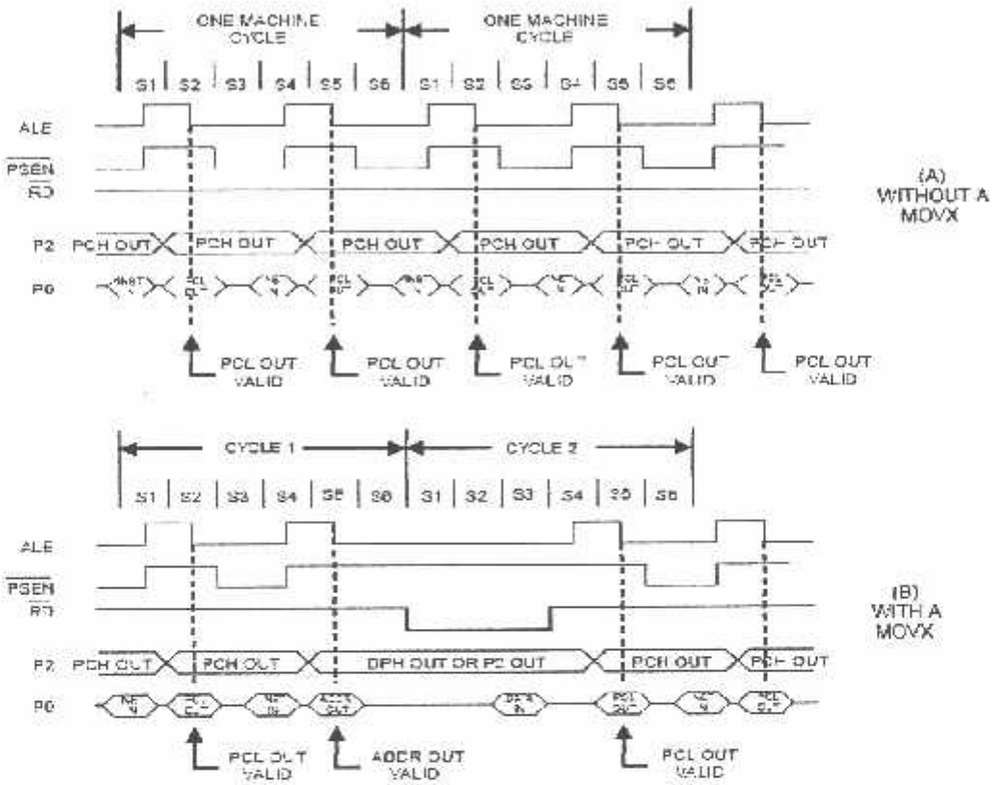
frekuensi oscillatornya sebesar 12 MHz. Setiap kondisi dibagi dalam Phase 1 half dan Phase 2 half.



Gambar 3.5 : Tahapan kondisi mikrokontroller AT89C51 [10]

Gambar di atas menunjukkan tahapan eksekusi pada kondisi-kondisi dan fase-fase untuk berbagai jenis perintah. Umumnya, dua program eksekusi dapat dijalankan selama setiap siklus, bahkan jika perintah yang sedang dijalankan tidak memerlukan hal itu. Jika perintah sedang dijalankan tidak memerlukan bit kode tambahan, maka CPU akan mengabaikan perintah tambahan dan Program Counter tidak ditambahkan.

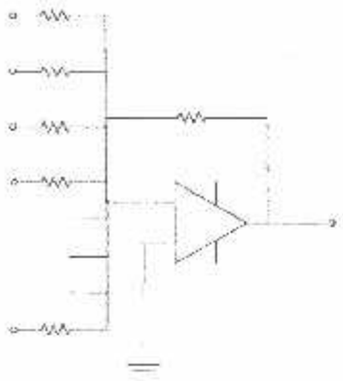
Eksekusi perintah satu siklus dimulai pada State 1 dari siklus mikrokontroller, ketika opcode dimasukkan ke dalam Instruction Register. Eksekusi kedua terjadi pada S4 di siklus yang sama. Eksekusi akan selesai pada akhir State 6 dari siklus ini.



Gambar 3.6 : Siklus yang dieksekusi dari program memory eksternal [10]

3.3. Perencanaan DAC

Sistem ini menggunakan DAC yang terdiri dari kumpulan resistor yang disusun menjadi bentuk ladder. Gambar rangkaian DAC N bit sederhana dapat dilihat pada gambar berikut :



Gambar 3.7 : Skema rangkaian DAC N bit [11]

Tegangan keluaran rangkaian DAC di atas adalah :

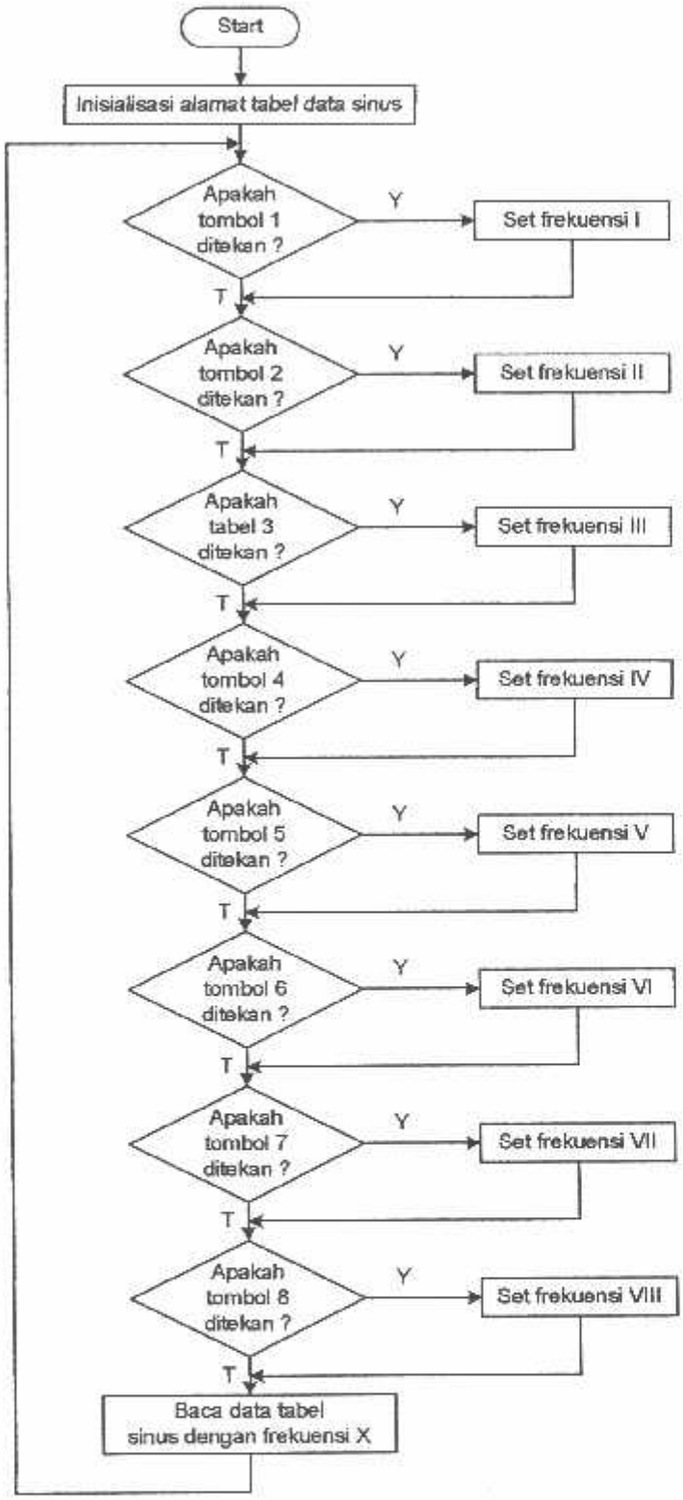
$$V_O = -R_f\left(\frac{b_1 V_{REF}}{R} - \frac{b_2 V_{REF}}{2R} + \frac{b_3 V_{REF}}{4R} + \ldots + \frac{b_N V_{REF}}{2^{N-1}R}\right) = -2\frac{R_f}{R}V_{REF}\left(\frac{b_1}{2} + \frac{b_2}{4} + \frac{b_3}{8} + \ldots + \frac{b_N}{2^N}\right)$$

Dengan b_1 sebagai MSB, b_N sebagai L_{SN}, dan V_{REF} adalah tegangan sinyal digital. Misal untuk perhitungan nilai tegangan analog berdasarkan data digital 8-bit adalah sebagai berikut:

$$V_{out} = V_{ref}\left[\frac{D_0}{256} + \frac{D_1}{128} + \frac{D_2}{64} + \frac{D_3}{32} + \frac{D_4}{16} + \frac{D_5}{8} + \frac{D_6}{4} + \frac{D_7}{2}\right]$$

3.3. Perencanaan Perangkat Lunak

Perangkat lunak ini berdasarkan pengendali utama yaitu mikrokontroller AT89C51. Pembuatan perangkat lunak sistem aplikasi berdasarkan pada semua proses yang harus dikerjakan perangkat keras. Perangkat lunak dibutuhkan oleh PC untuk memberikan perintah kepada mikrokontroller. Bahasa pemrograman yang digunakan adalah bahasa Delphi. Pembuatan perangkat lunak harus melalui proses-proses uji coba secara *software* maupun secara *hardware*.



Gambar 3.8 : Flow chart mikrokontroller untuk DDS

BAB IV

PENGUJIAN SISTEM

Pengujian dan pengukuran dalam hal ini dilakukan dengan tujuan untuk memastikan bahwa *Function generator* dapat bekerja sesuai dengan perencanaan. Bab ini menguraikan tentang bagian alat yang diuji, tujuan pengujian, langkah-langkah pengujian dan hasil pengujian yang menunjukkan unjuk kerja dari tiap-tiap bagian alat.

Pembahasan dalam bab ini dibagi menurut pembagian alat yang diuji untuk mengetahui unjuk kerja sistem secara keseluruhan. Untuk mengetahui kemampuan alat dan sistem kerja sesuai dengan program yang telah dibuat maka dilakukan pengujian pada alat dan sistem kerja alat.

Pengujian dilakukan pada tiap-tiap blok sistem adapun blok-blok yang di uji adalah:

1. Rangkaian DAC R2R
2. Rangkaian *Function Generator*

4.1. Pengujian Rangkaian DAC R2R

a. Tujuan

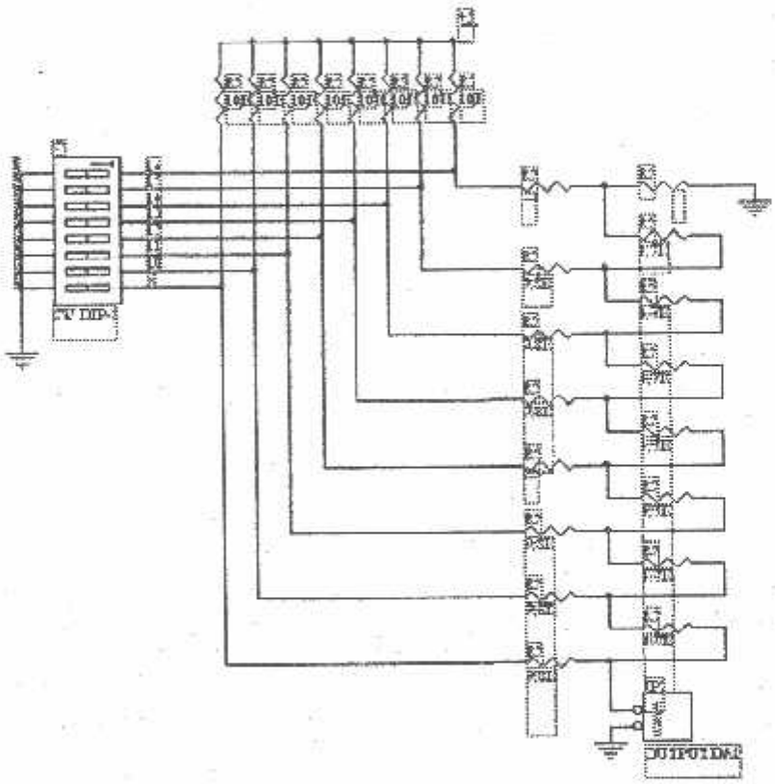
Penggujian rangkaian DAC bertujuan untuk mengetahui apakah DAC berfungsi dengan baik sekaligus untuk mengetahui kelancaran proses konversi atau perubahan sinyal digital ke sinyal analog

b. Peralatan yang digunakan

- Rangkaian DAC R2R
- Sumber tegangan +12 Volt
- Switch 8 Buah
- Multimeter digital (DT9205B)

c. Langkah Pengujian

1. Merangkai rangkaian seperti pada gambar dibawah ini:



Gambar 4.1 : Rangkaian Pengujian DAC R2R

2. Menghubungkan kutub positif multimeter digital (multimeter diset pada skala 20 Volt DC) pada outputan DAC, sedangkan kutub negative ke ground.
3. Mengamati perubahan nilai tegangan pada multimeter dan mencatat hasilnya pada tabel 4-1.

Untuk mengetahui keluaran DAC R2R dapat dicari dengan menggunakan rumus berikut:

$$V_{out} = V_{ref}\left[\frac{D_0}{256} + \frac{D_1}{128} + \frac{D_2}{64} + \frac{D_3}{32} + \frac{D_4}{16} + \frac{D_5}{8} + \frac{D_6}{4} + \frac{D_7}{2}\right]$$

Dimana:

V_{ref} = 5 Volt

Jika diketahui inputan= 00000001_B. Maka keluaran DAC R2R :

$$V_{out} = V_{ref}\left[\frac{D_7}{2} + \frac{D_6}{4} + \frac{D_5}{8} + \frac{D_4}{16} + \frac{D_3}{32} + \frac{D_2}{64} + \frac{D_1}{128} + \frac{D_0}{256}\right]$$

$$V_{out} = 5\left[\frac{0}{2} + \frac{0}{4} + \frac{0}{8} + \frac{0}{16} + \frac{0}{32} + \frac{0}{64} + \frac{0}{128} + \frac{1}{256}\right]$$

= 19.53 mV

Jika diketahui inputan= 11111111_B. Maka keluaran DAC R2R :

$$V_{out} = V_{ref}\left[\frac{D_7}{2} + \frac{D_6}{4} + \frac{D_5}{8} + \frac{D_4}{16} + \frac{D_3}{32} + \frac{D_2}{64} + \frac{D_1}{128} + \frac{D_0}{256}\right]$$

$$V_{out} = 5\left[\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \frac{1}{32} + \frac{1}{64} + \frac{1}{128} + \frac{1}{256}\right]$$

= 4.95 Volt

Berikut merupakan tabel perbandingan tegangan output antara hasil pengukuran dan hasil perhitungan dari keluaran DAC R2R :

Tabel 4.1: Data Pengujian Rangkaian Konversi DAC R2R

NO	Bit Masukan	Keluaran DAC R2R Tegangan mV (mili Volt)		
		Perhitungan	Pengukuran	% kesalahan
1	00000001	19.53	19.5	$1.53 . 10^{-3}$
2	00000010	39.06	33.9	0.132
3	00000011	58.59	51.4	0.122
4	00000100	78.125	66.9	0.143
9	11111111	4.98 (V)	4.95 (V)	$6.02 .10^{-3}$

Untuk % kesalahan dapat dihitung dari hasil pengukuran dan perhitungan sebagai berikut :

$$\%\text{kesalahan} = \frac{(perhitungan - pengukuran)}{perhitungan} \times 100\%$$

Contoh :

V_0 perhitungan = 19.53

V_0 pengukuran = 19.5 Volt

$$\%\text{ kesalahan} = \frac{(19.53) - (19.5)}{(19.53)} \times 100\%$$

jadi % kesalahan = $1.53 . 10^{-3}$

Dari tabel di atas didapatkan error rata–rata :

$$= \frac{(1.53.10^{-3} + 0.132 + 0.122 + 0.143 + 6.02.10^{-3})\%}{5}$$

$$= 0.08\ \%$$

4.2. Pengujian Rangkaian *Function Generator*

a. Tujuan

Untuk mengetahui bentuk sinyal dan besar frekuensi yang dibangkitkan oleh rangkaian *Function Generator* apakah telah sesuai perencanaan.

b. Peralatan yang digunakan

- Rangkaian *Function Generator*
- *Oscilloscope* (Tektronix TDS 210)

c. Langkah pengujian

Pengujian dilakukan dengan cara melihat bentuk gelombang *output* dan nilai frekuensi yang dikeluarkan *Function Generator* yang telah dibuat pada *osciloscope* pada pin *output*. Pengujian dilakukan berkali-kali, pada nilai frekuensi dan bentuk gelombang yang berbeda. *Osiloscope* diset pada 2 V/div, Probe merah pada *Osiloscope* dipasang pada output dan probe hitam *Osiloscope* dipasang pada *ground*.

d. Hasil pengujian

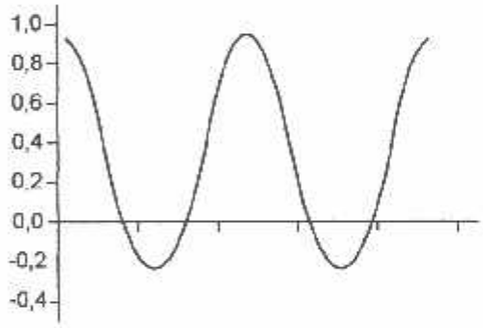
Dalam hal ini dilakukan pengujian terhadap sistem yang dibuat. Tujuan pengujian ini adalah untuk mengetahui kinerja sistem tersebut. Rangkaian elektronik yang diuji dalam hal ini adalah sebuah amplifier jenis operasional.

Pengujian dilakukan untuk mengetahui kinerja sistem yang telah dirancang dan dibuat. Sebelum dilakukan pengujian maka dibuat grafik sinus berdasarkan perhitungan dengan Microsoft Excel. Input untuk grafik sinus :

Tabel 4.2 : Input grafik sinus

DB	100	112	124	136	147	DB	158	168	176	184	190
DB	194	198	199	199	198	DB	195	190	185	177	169
DB	159	149	138	126	114	DB	101	90	77	65	54
DB	43	34	25	17	11	DB	6	3	1	1	2
DB	5	9	15	22	30	DB	39	50	61	73	85

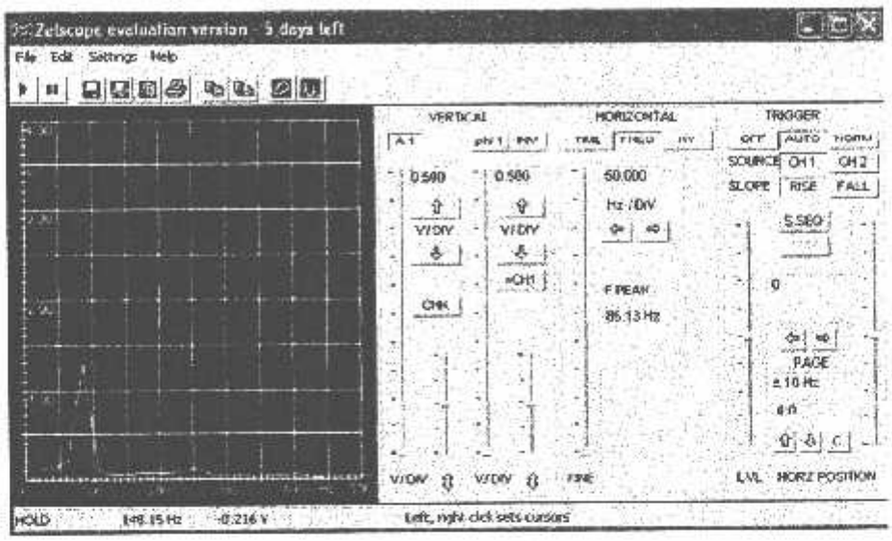
Didapat plot grafik sinus sebagai berikut :



Gambar 4.2 : Grafik sinus teoritis

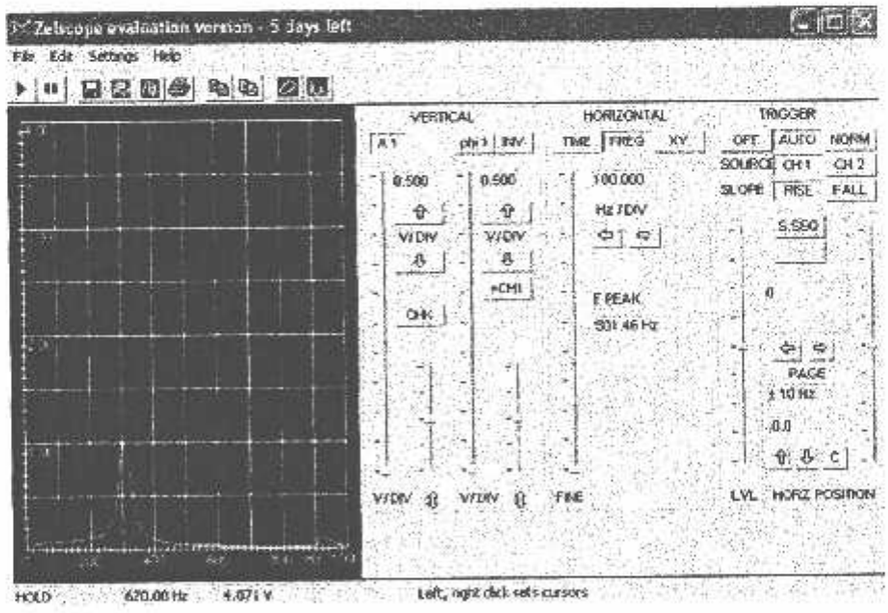
4.3. Output Frekuensi

a. Frekuensi 86 Hz



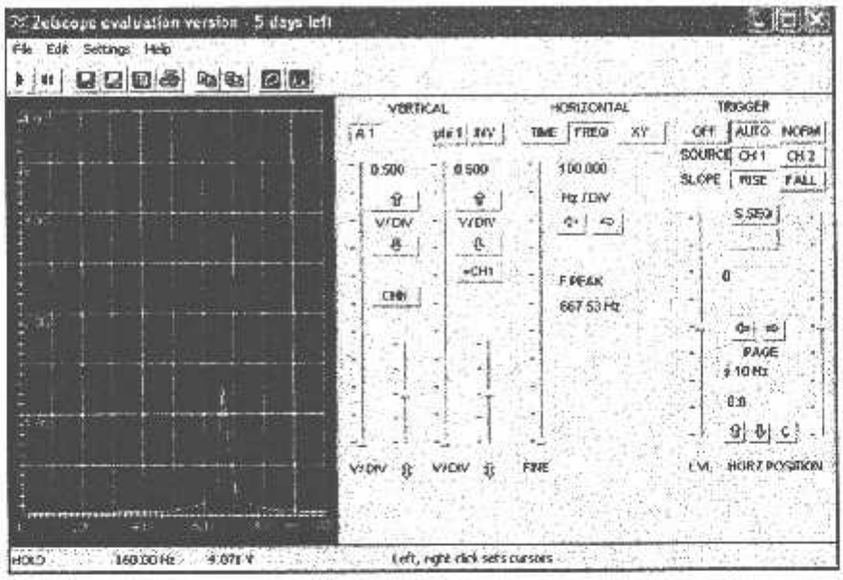
Gambar 4.3 : Output 86 Hz

b. Frekuensi 300 Hz



Gambar 4.4 : Output 300 Hz

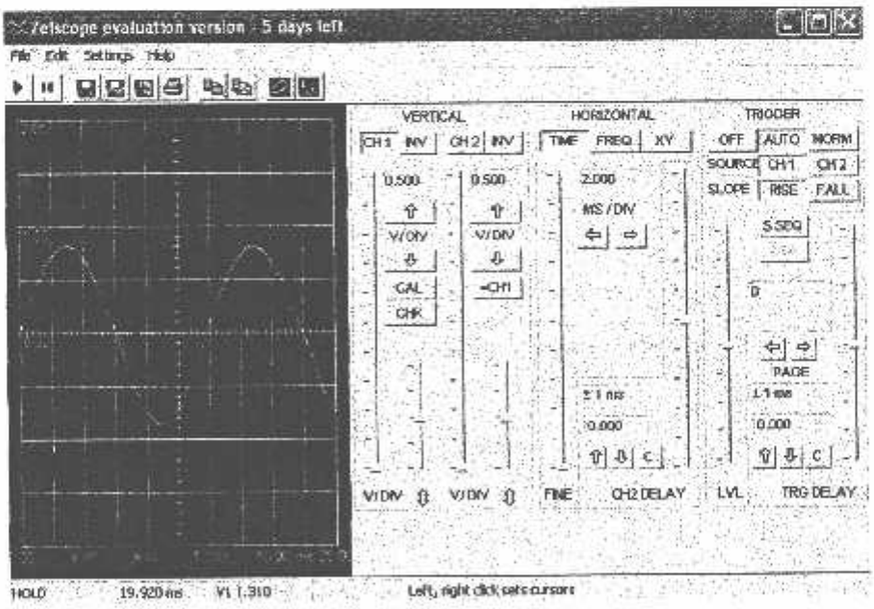
c. Frekuensi 600 Hz



Gambar 4.5 : Output 600 Hz

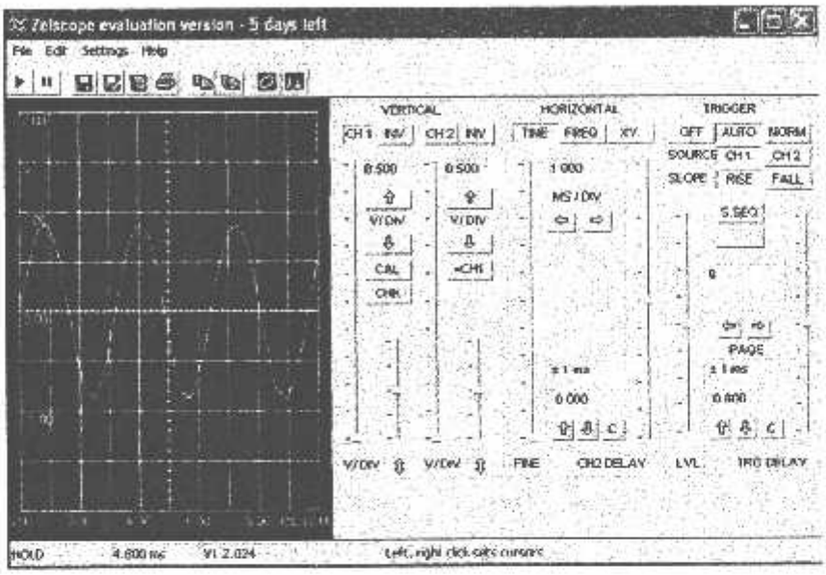
4.4. Output Waktu

a. Waktu rendah



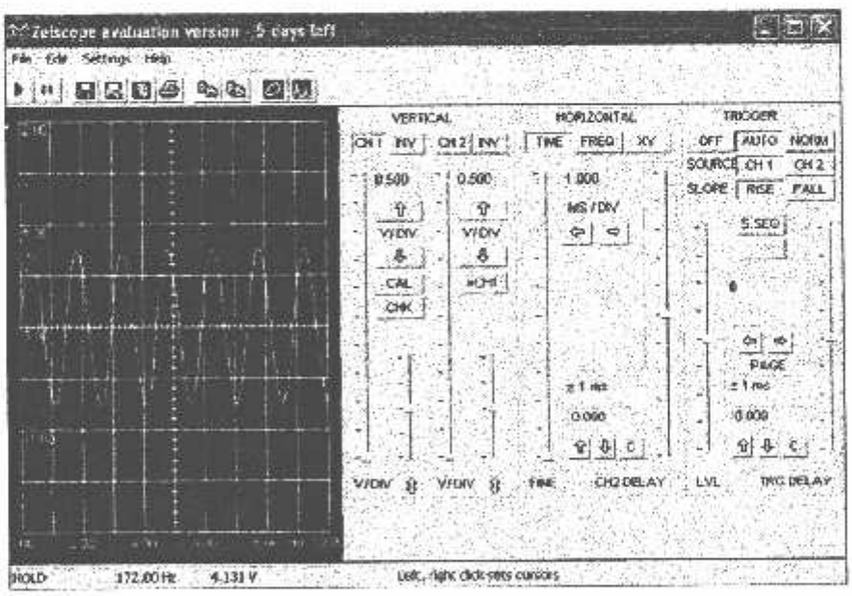
Gambar 4.6 : Output waktu rendah

b. Waktu sedang



Gambar 4.7: Output waktu sedang

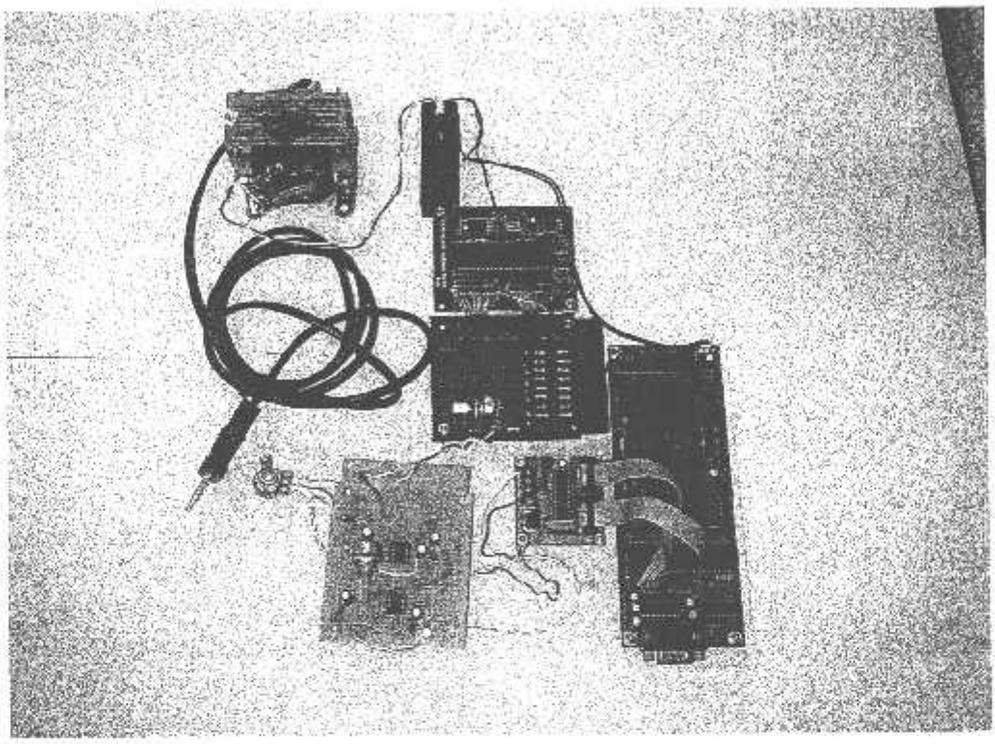
c. Waktu tinggi



Gambar 4.8 : Output waktu tinggi

4.5. Analisa

Hasil perancangan sistem dibuat dalam bentuk set peralatan sebagai berikut :



Gambar 4.9 : Foto alat

Hasil pengujian menunjukkan bahwa tampilan di komputer telah menunjukkan pola grafik sinus. Hal ini berarti bahwa sistem telah dapat menghasilkan grafik sinus sesuai dengan grafik hasil perhitungan. Tetapi sistem ini masih memerlukan penyempurnaan karena tampilan grafik sinus di atas masih harus diteliti lagi kesesuaian frekuensinya. Selain itu hasil tampilan grafik sinus masih belum sempurna karena tidak melalui proses filter.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Dari hasil perancangan dan pengujian sistem yang telah dilakukan, dapat diambil kesimpulan sebagai berikut :

1. Rangkaian DAC sudah akurat, tetapi hasil pengukuran menunjukkan tingkat error rata-rata 0,08%.
2. Dari hasil pengujian alat, bentuk gelombang sinusoidal masih belum sempurna karena tidak melalui proses filter.
3. Karena keterbatasan mikrokontroller AT89C51, sistem ini hanya akurat untuk frekuensi tertentu.
4. Generator sinyal untuk sistem penguji rangkaian elektronik berbasis mikrokontroller AT89C51 telah berfungsi tetapi masih memerlukan penyempurnaan lebih lanjut.

5.2. Saran

Perlu dilakukan perbaikan sistem menyangkut kesesuaian frekuensi.

DAFTAR PUSTAKA

[1]. Analog Devices.Inc, 2008, **All About Direct Digital Synthesis**

[2]. Dattalo S, 2008, Sine Waves

[3]. Futurlec, 2009, **AT89C51 - 8 bit Microcontroller with 4k Bytes Flash**

[4]. Manikanandan K, 2000, **Direct Digital Synthesis Part-I**, Electronic Technology

[5]. National Instrumen Corporation, 2006

[6]. NationMaster.com, 2003, **Signal Generator**

[7]. Stanford Research System, **Direct Digital Synthesis**, www.thinkSRS.com

[8]. Suyono, 2007, **Panduan Praktis Mikrokontroller AT89C51**, IlmuKomputer.com

[9]. http://www.atmel.com

[10]. Vastianos G, 1998, **10Hz - 100KHz D.D.S. Function Generator**, Technological Educational Institute of Piraeus, Greece

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1
Jl. Raya Karanglo Km.2
Malang

Formulir Bimbingan Skripsi

Nama : **DIDI AFFANDI**
Nim : **98.17.006**
Masa Bimbingan : **20 Desember 2008 – 20 Juni 2009**
Judul : **PERANCANGAN DAN PEMBUATAN GENERATOR SINYAL UNTUK SISTEM PENGUJI RANGKAIAN ELEKTRONIK BERBASIS MICROKONTROLLER AT89C51**

No	Tanggal	Uraian	Paraf
1		Bab I → IV num.	
2		Bab II	
3		Bab III	
4		Bab IV	
5		Lengkapi bab	
6		Lapran Langsung	

Malang, 10 Maret 2010
Dosen Pembimbing

Ir.F.Yudi Limpraptono, MT
NIP Y. 1039500274

FORM S-4B

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Janjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika / T.Infokom, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : DIDI
NIM : 98.17.006
Perbaikan meliputi :

·) BAB III
- DESAIN TABEL → DARI MANA ??
- EFISIENSI yg dimaksudkan (DIHITUNG)

·) KESIMPULAN
·) SARAN (DIBANDINGKAT dgn BAB III)

Malang, 200

( )

1

```
        ORG OH
        MOV 30H, #100

SINUS : MOV DPTR, #SINE
        MOV RO, #51

ULANG : CLR A
        MOVC A, OA+DPTR
        MOVC P1,A
        INC DPTR
        JB P3.O, CEK
        MOV 30H, #100

CEK :   JB P3.1, CEK1
        MOV 30H, #50

CEK1 : JB P3.2, CEK2
        MOV 30H, #25

CEK2 : JB P3.3, CEK3
        MOV 30H, #15

CEK3 : JB P3.4, CEK4
        MOV 30H, #10

CEK4 : JB P3.5, CEK5
        MOV 30H, #5

CEK5 : JB P3.6, CEK6
        MOV 30H, #1

CEK6 : JB P3.7, TERUS
        MOV 30H, #0

TERUS : MOV A, 30H
        CJNE A, #0, BABLAS
        JMP LEWAT

BABLAS : CALL TUNDA

LEWAT : DJNZ RO, ULANG
        JMP SINUS

TUNDA : MOV R1, 30H
        DJNZ R1, $
        RET
```

```
SINE : DB 100,112,124,136,147,158,168,176,185,190
       DB 194,198,199,199,198,195,190,185,177,169
       DB 159,149,138,126,114,101,090,077,065,054
       DB 043,034,025,017,011,006,003,001,001,002
       DB 005,009,015,022,030,039,050,061,073,085,097

END
```

DIRECT DIGITAL SYNTHESIS PART-I

MANIKANANDAN K.

In direct digital synthesis (DDS) technique, signal is generated in the form of a series of digital numbers and is converted into analogue by using a digital-to-analogue converter (DAC). The output of the system is directly proportional to the frequency-setting word

Recent years have witnessed an increasing interest in direct digital synthesis (DDS). The technique emerged in the early 80s and was probably known in principle for several years before that. In those early days, the highest output frequency attainable was no more than a few MHz, limited by the performance of the then available logic ICs and digital-to-analogue converters (DACs).

With the advancement in semiconductor technology, the attainable performance has increased by leaps and bounds. DDS chips realised in gallium arsenide now provide output frequencies of over 400 MHz. Silicon technology delivers output frequencies in excess of 300 MHz. However, when operating at the top of the frequency range, the worst case levels of spurious outputs have not changed much, still being in the region of 40 to 45 dB down on the wanted output.

RF carrier generation

The long-term frequency stability of an LC oscillator is generally poor. It is difficult to construct a small inductor with stability much better than 0.01 per cent. Its spectral purity can only be increased by increasing the operating Q of the tuned circuit, which again is not easy, especially in the case of miniaturised equipment.

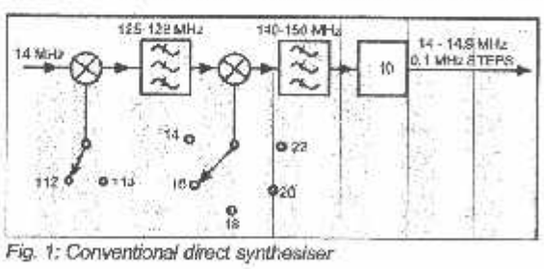


Fig. 1: Conventional direct synthesiser

In fixed-frequency applications, the situation is much rosier. A crystal resonator can have a working Q of the order of 10^5 to 10^6, making it possible to design oscillators with very low sideband noise and excellent sideband and long-term frequency stability, especially if the crystal is maintained at a constant temperature in an oven.

Direct synthesis

The ideal carrier generator would be the one with the tuning capacity of an LC oscillator and the stability of a crystal oscillator—and the next major advance in the carrier technology provided just this. With two crystals, at different frequencies f1 and f2, one can obtain four different frequencies by mixing them, so as to provide four distinct frequencies f1, f2, (f1+f2), and (f1-f2). If one uses harmonics or sub-harmonics of f1 and f2 as well, the possibilities rapidly increase. This principle can be extended to obtain any frequency with any degree of resolution.

A typical block diagram of the system used is shown in Fig. 1. The system is called the direct synthesiser, since the output frequency is ultimately obtained from master crystal oscillator and has almost the purity of the master. The mix\filter\divide block provides 100kHz steps. Using the output as the input to another, similar block provides an output with 10kHz steps. Further blocks can be added to provide any required degree of resolution. High-performance filters and careful screening within and between blocks enable all non-harmonic spurious to be held at 80 dB down on the wanted output, at a price. The term 'direct analogue synthesiser' is sometimes used to distinguish it from the direct digital synthesiser.

Direct synthesis is an attractive route in applications where high purity, low spurious outputs, and fast switching are required, especially in lab and ATE equipment. But the size and cost make it difficult to apply in portable military or OEM equipment.

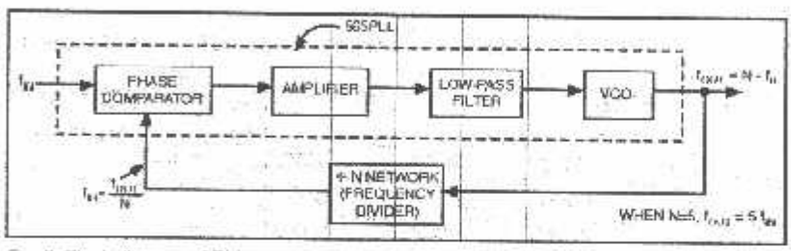


Fig. 2: Block diagram of PLL system

Phase lock loop

The PLL (phase lock loop) technique is widely applied in space applications such as synchronising a receiver to a spacecraft signal with a doppler shift.

Fig. 2 shows the block diagram of a PLL system for producing an output of N times the reference frequency, which

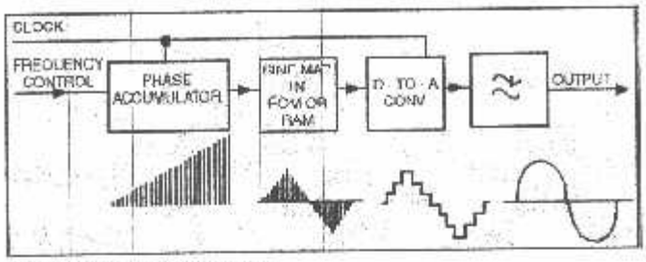


Fig. 3: Direct digital synthesiser

may itself be obtained from a crystal oscillator. When the loop is in lock, the output of the 'divide by N' stage is at exactly the same frequency as the reference, but there will be a small phase offset, leading or lagging, resulting in a DC component in the output of the mixer. This output is applied via a low-pass filter to a voltage-controlled oscillator and results in exactly the desired output frequency.

If N is set to a different value, the output frequency will change; if N is changed to N+2, the output frequency will increase by twice the reference frequency. Here the step size is equal to the reference frequency. Enhancements of this technique, such as dual-modulus prescalars and multiple-loop architecture, enable fine frequency-resolution to be obtained comparatively at low cost. A 'fractional N' approach enables the performance-to-price ratio to be improved even further, making the PLL synthesiser currently the most commonly encountered scheme. PLL synthesisers offer fine frequency-resolution and very low levels of spurious outputs at comparatively low cost, but cannot achieve the very same low levels of close in phase noise as the direct synthesiser. These have slow switching times, due to loop-filter settling time.

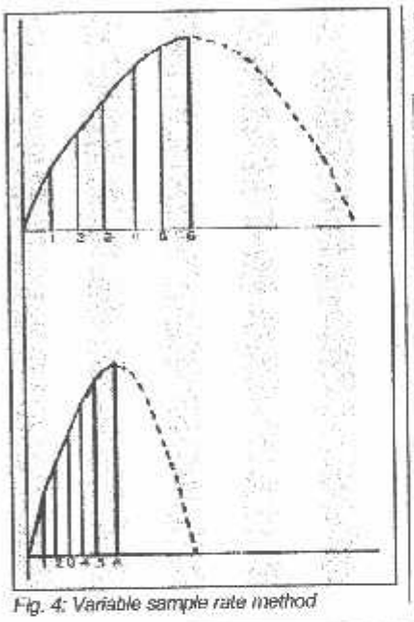


Fig. 4: Variable sample rate method

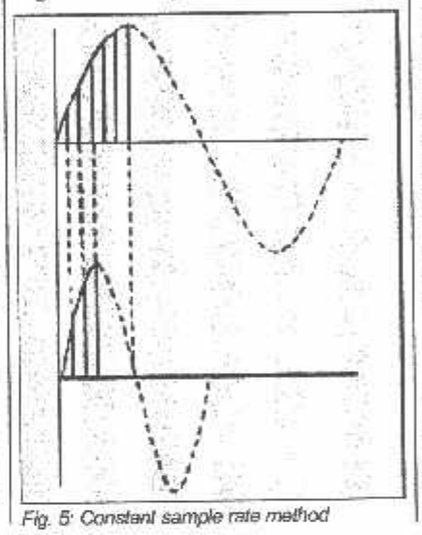


Fig. 5: Constant sample rate method

Direct digital synthesis

With the exception of the output low-pass filter, the whole scheme is entirely digital and is eminently suited for large-scale integration into a single chip. The output of the system is directly propor-

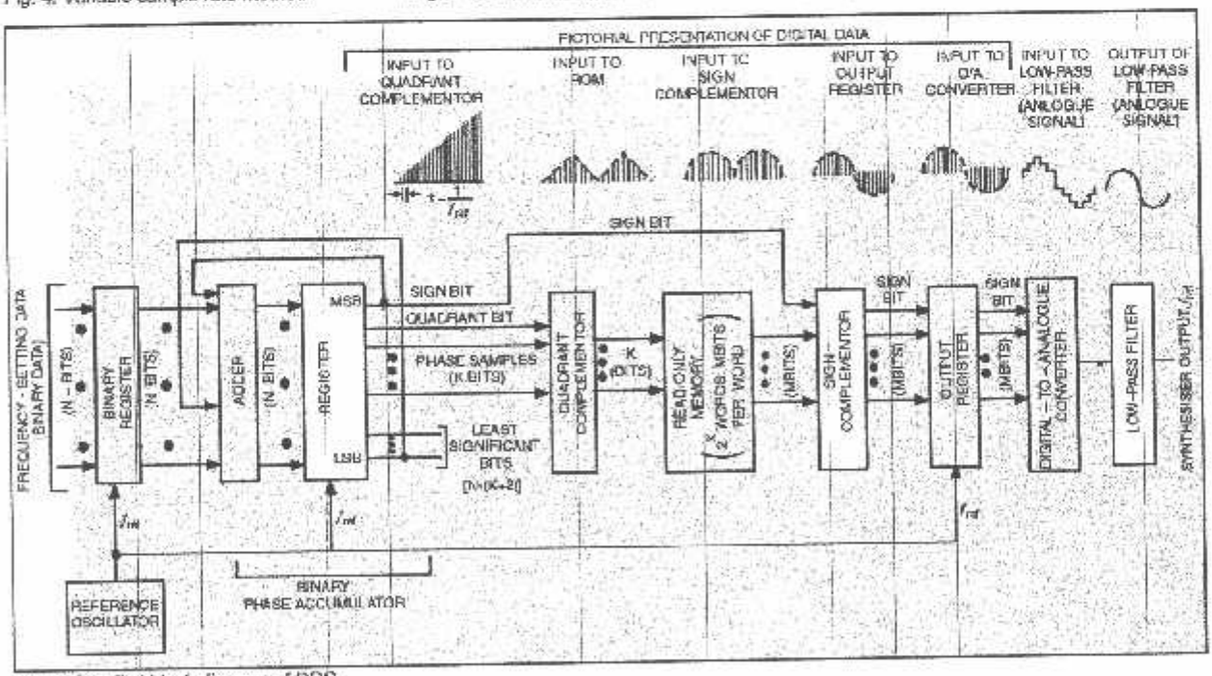


Fig. 6: Detailed block diagram of DDS

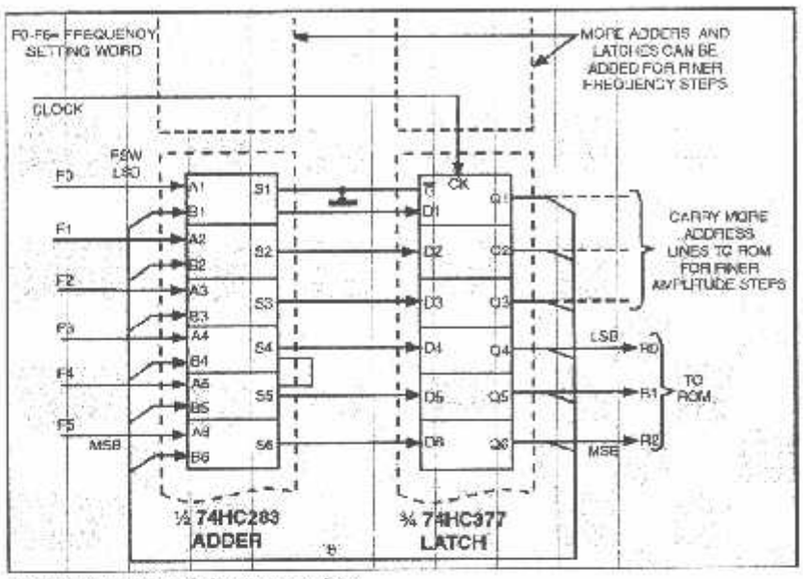


Fig. 7: The working of phase accumulator

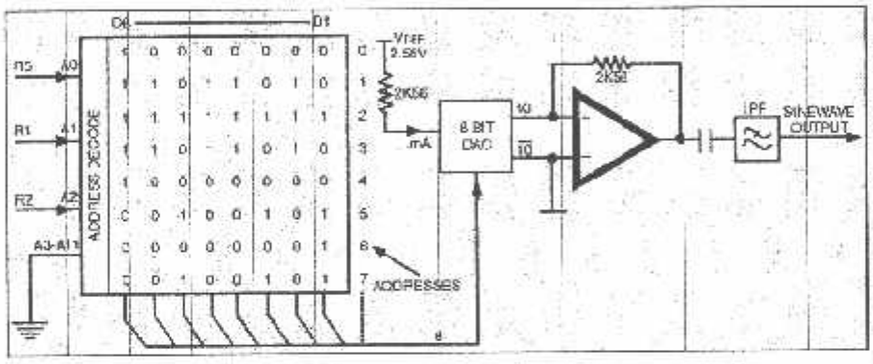


Fig. 8: The rest of the rudimentary DDS

tional to the frequency-setting word.

In this technique, a signal is generated in the form of a series of digital numbers and is converted into analogue by using a DAC (Fig. 3). These numbers are stored in a memory device known as the memory lookup table.

One way of generating the output is to output some points for every cycle of the sinewave. In this case, tuning is effected by changing the rate at which the data is output (refer Fig. 4). It is difficult to get fine frequency-resolution using this method. In the second case, the rate at which data is output is kept constant. The basic idea is to store N uniformly spaced samples in the memory. The lowest output frequency contains N distinct samples. Now if one outputs every other point stored in the ROM at the same rate, one gets an output with twice the original frequency. So, if every Kth sample is taken, a K times faster waveform will be obtained. The frequency resolution is same as that of the lowest frequency. This method is illustrated in Fig. 5.

A more detailed block diagram of the DDS is shown in Fig. 6. The frequency setting word is given to the adder of a phase accumulator through a binary register. The adder adds the digital signals from the binary register to the value of the accumulator register and updates the accumulator register with the most recent sum. This register transfers the digital data from the output of the adder to its input at every clockpulse, so as to make the accumulator overflow at regular intervals. The accumulator now addresses the ROM through a quadrant complimentor.

Since all the amplitude information for a full 360° sinewave is contained in 90° of information, only 90° of memory mapping is required. The quadrant complimentor simply helps the accumulator to clock in both the directions, thus giving 180° of the wave. Now the sine complimentor changes sign and the process is repeated to get the full sinewave.

The phase accumulator

Fig. 7 shows how the phase accumulator works. For simplicity, a 6-bit accumulator with three most significant bits controlling the read only memory with just eight memory locations has been assumed. The output of the three most significant bits from the latch to the ROM needs to represent one complete cycle of the output waveform, so the eight values in the ROM need to represent the amplitude of the sinewave at 45° intervals. The higher the value of the frequency setting word, the fewer cycles will be needed to cycle through the ROM once, and hence the higher output frequency. The lowest possible output frequency results when the frequency-setting word is 000001. In this case, every eighth clock pulse advances the ROM address by one.

The lookup table

Fig. 8 shows the rest of the rudimentary system, ROM, DAC, and output fil-

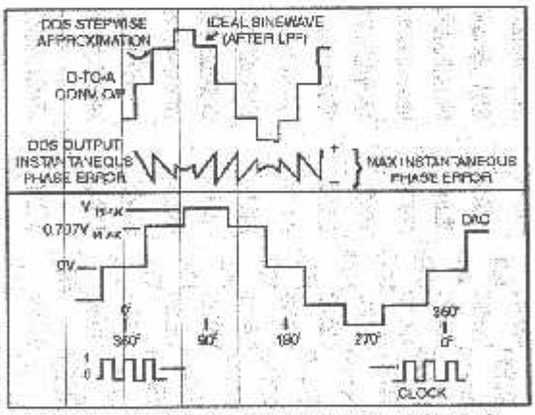


Fig. 9: 8Hz and 4Hz outputs from DDS, with a 64Hz clock

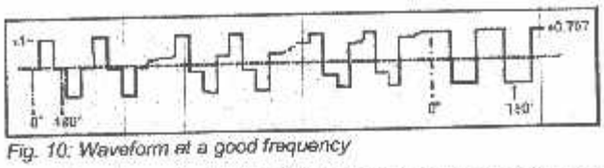


Fig. 10: Waveform at a good frequency

ter. The values stored in ROM range from 1 representing the negative peak of the sinewave, to 255 representing the positive peak and definite points on a sinewave in offset binary. The value 128 corresponds to the mean level (0V) of the waveform.

Operation

Imagine a 6-bit latch clocked at 64 Hz and the accumulator loaded with 000001 (Fig. 7). Whatever the current value held in the latch, the output of the adder will exceed by 1. So, in the next clock pulse, the new value will be loaded into the latch, and so on repeatedly. Starting from the stage where the latch content is all zero, the three ROM address lines will be zero. The ROM will output 10000000, causing the DAC output to be 1.28V, corresponding to the zero value of sinewave at 0°. The output of the LPF will indeed be 0V, due to the blocking capacitor between it and the opamp; the purpose of the latter being to turn the DAC's current output back into a voltage.

Eight clock cycles later, the accumulator contents will be 001000; Ro will now be 1 so that the ROM output is 11011010, corresponding to the binary offset representation of the value of a sinewave at 45°, and accordingly the DAC output will becomes 2.18V.

After 64 clock cycles, the accumulator is full and the DAC has delivered a complete cycle of stepwise representation of a sinewave. Thus the lowest non-zero output frequency is 1 Hz or generally Fclk/2^N, where N is the number of the accumulator bits (Fig. 9). If the frequency setting word is set to 000010, the whole process will complete one output cycle in half the clock period. So, Fclk/2^N is also the smallest frequency increment available.

Fig. 4 also shows the instantaneous departure of the stepwise output approximation from a continuous sinewave. These phase perturbations occur at a rate equal to the clock frequency, which is well above the highest output frequency, and so these can be suppressed by LPF even if one has a perfect DAC with infinite resolution. Amplitude truncation errors cause sinewave representations to be imperfect, resulting in harmonics in the output; whether these appear in the output depends on commanded output frequency. Apart from this possible harmonic content, the output of 6-bit DDS is perfect at frequency setting of 1, 2, 4, and 8 Hz. At other frequencies, the operation becomes a little more complicated.

There are many other mechanisms that produce unwanted, non-harmonic, spurious outputs. When the demanded frequency is so high that the ROM only outputs a subset of its contents on each cycle, the waveform into the LPF can take various quite different forms from a sinewave. Fig. 10 shows some shapes the waveform can take at a good frequency where the ROM address is incrementing so as to output just four equally spaced locations.

To be continued...

XILINX® LogiCORE

DDS v5.0

S246 April 28, 2005

Product Specification

eatures

Drop-in module for Virtex™, Virtex-E, Virtex-II, Virtex-II Pro, Virtex-4, Spartan™-II, Spartan-IIE, Spartan-3, and Spartan-3E FPGAs

Sine, Cosine, or quadrature outputs

Look-up table can be allocated to distributed or block memory

Phase dithering and Taylor series correction option provide high dynamic range signals using minimal amount of FPGA resources. SFDR range is 18 dB to 115 dB

Phase dithering removes the spectral line structure associated with conventional phase truncation waveform synthesis architectures

Support for 1 to 16 independent channels

High-precision synthesizor with fine frequency resolution (Δf = 0.02 Hz @ fclk = 100 MHz, 32-bit phase accumulator)

4 to 32-bit two's complement output sample precision

Optional phase offset capability providing support for multiple synthesizers with precisely controlled phase differences

Simple fixed-output frequency option

Uses relationally placed macro (RPM) mapping and placement technology for maximum and predictable performance

Incorporates Xilinx Smart-IP™ technology for utmost parameterization and optimum implementation

- For use with v7.1i and later of the Xilinx CORE Generator™ system

Applications

- Digital radios and modems
- Software-defined radios (SDR)
- Digital down/up convertors for cellular and PCS base stations
- Waveform synthesis in digital phase locked loops
- Generating injection frequencies for analog mixers

General Description

Direct digital synthesizers (DDS), or numerically controlled oscillators (NCO), are important components in many digital communication systems. Quadrature synthesizers are used for constructing digital down and up converters, demodulators, and implementing various types of modulation schemes, including PSK (phase shift keying), FSK (frequency shift keying), and MSK (minimum shift keying). A common method for digitally generating a complex or real valued sinusoid employs a look-up table scheme. The look-up table stores samples of a sinusoid. A digital integrator is used to generate a suitable phase argument that is mapped by the look-up table to the desired output waveform. A simple user interface accepts system-level parameters such as the desired output frequency and spur suppression of the generated waveforms.

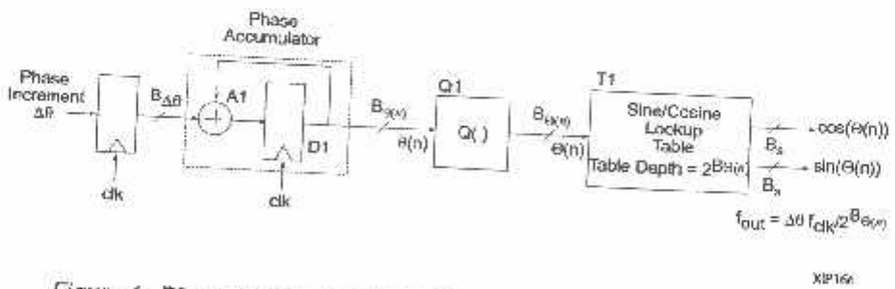


Figure 1: **Phase Truncation DDS (A simplified view of the DDS core)**

'heory of Operation

, high-level view of the DDS Core is presented in Figure 1. 'he integrator (components D1 and A1) computes a phase lope that is mapped to a sinusoid (possibly complex) by the)ok-up table T1. The quantizer Q1, which is simply a slicer, ccepts the high-precision phase angle $\Theta(n)$ and gener-tes a lower precision representation of the angle denoted s $\Theta(n)$ in the figure. This value is presented to the address ort of a look-up table that performs the mapping from hase-space to time.

he fidelity of a signal formed by recalling samples of a nusoid from a look-up table is affected by both the phase nd amplitude quantization of the process. The length and idth of the look-up table affect the signal's phase angle esolution and the signal's amplitude resolution respec-vely. These resolution limits are equivalent to time base jit-ir and to amplitude quantization of the signal, and add pectral modulation lines and a white broad-band noise oor to the signal's spectrum.

irect digital synthesizers use an addressing scheme with 1 appropriate look-up table to form samples of an arbitrary equency sinusoid. If an analog output is required, the DDS 'esents these samples to a digital-to-analog converter)AC) and a low-pass filter to obtain an analog waveform ith the specific frequency structure. Of course, the sam-es are also commonly used directly in the digital domain. he look-up table traditionally stores uniformly spaced sam-es of a cosine and a sine wave. These samples represent single cycle of a length $N = 2^{B_{\Theta(n)}}$ prototype complex sinu-oid and correspond to specific values of the sinusoid's gument $\Theta(n)$ as shown in Eq.(1).

$$\Theta(n) = n\frac{2\pi}{N} \tag{1}$$

here n is the time series sample index.

uarter wave symmetry in the basis waveform can be ploited to construct a DDS that uses shortened tables. In is case, the two most significant bits of the quantized ase angle $\Theta(n)$ are used to perform quadrant mapping. is implementation results in a more area efficient imple-entation because the memory requirements are mini-ized: either fewer FPGA block RAMs or reduced stributed memory. Based on the Core customization rameters, the DDS core will automatically employ quar-r-wave symmetry when appropriate[1].

utput Frequency

ie output frequency, f_{out}, of the DDS waveform is a func-n of the system clock frequency f_{clk}, the number of bits $_{\Theta(n)}$ in the phase accumulator and the phase increment lue $\Delta\theta$. That is, $f_{out} = f(f_{clk}, B_{\Theta(n)}, \Delta\theta)$.

Output frequency in Hertz is defined as

$$f_{out} = \frac{f_{clk}\Delta\theta}{2^{B_{\Theta(n)}}} \text{ Hz} \tag{2}$$

For example, if the DDS parameters are

$$\begin{aligned} f_{clk} &= 120 \text{ MHz} \\ B_{\Theta(n)} &= 10 \\ \Delta\theta &= 12_{10} \end{aligned} \tag{3}$$

the output frequency will be

$$\begin{aligned} f_{out} &= \frac{f_{clk}\Delta\theta}{2^{B_{\Theta(n)}}} \text{ Hz} \\ &= \frac{120\times10^6\times12}{2^{10}} \\ &= 1406250 \text{ MHz} \end{aligned} \tag{4}$$

The phase increment value required to generate an output frequency f_{out} Hz is

$$\Delta\theta = \frac{f_{out}2^{B_{\Theta(n)}}}{f_{clk}} \tag{5}$$

Frequency Resolution

The frequency resolution Δf of the synthesizer is a function of the clock frequency and the number of bits $B_{\Theta(n)}$ employed in the phase accumulator. The frequency resolution can be determined using the following equation

$$\Delta f = \frac{f_{clk}}{2^{B_{\Theta(n)}}} \tag{6}$$

For example, for the DDS parameters

$$\begin{aligned} f_{clk} &= 120 \text{ MHz} \\ B_{\Theta(n)} &= 32 \end{aligned} \tag{7}$$

the frequency resolution is

$$\begin{aligned} \Delta f &= \frac{f_{clk}}{2^{B_{\Theta(n)}}} \\ &= \frac{120\times10^6}{2^{32}} \\ &= 0.0279396 \text{ Hz} \end{aligned} \tag{8}$$

Phase Increment

The phase increment is an unsigned value. The phase increment term $\Delta\theta$ defines the synthesizer output frequency. Consider a DDS with the following parameterization

For very short tables, FPGA logic resources are actually minimized by storing a complete cycle. The user is not required to make any design decisions in this context; the CORE Generator will always produce the smallest core possible.

$$f_{clk} = 100 \text{ MHz}$$
$$B_{\theta(n)} = 28 \qquad (9)$$
$$B_{\Theta(n)} = 12$$

o generate a sinusoid with frequency $f_{out} = 19$ MHz, the equired phase increment would be

$$\Delta\theta = \frac{f_{out} 2^{B_{\Theta(n)}}}{f_{clk}} = \frac{19 \times 10^6 \times 2^{12}}{100 \times 10^6} = 778.24 \qquad (10)$$

pectral Purity Considerations

he fidelity of a signal formed by recalling samples of a nusoid from a look-up table is affected by both the phase nd amplitude quantization of the process. The length and idth of the look-up table affect the signal's phase angle esolution and the signal's amplitude resolution respecvely. These resolution limits are equivalent to time base jitr and to amplitude quantization of the signal and add pectral modulation lines and a white broad-band noise oor to the signal's spectrum.

n conjunction with the system clock frequency, the phase ccumulator width determines the frequency resolution of ie DDS. The accumulator must have a sufficient field width span the desired frequency resolution. For most practical pplications, a large number of bits are allocated to the hase accumulator in order to satisfy the system frequency esolution requirements. By way of example, if the required esolution is 1 Hz, and the clock frequency is 100 MHz, the equired field width of the accumulator is

$$B_{\theta(n)} = \log_2\left\lceil \frac{f_{clk}}{\Delta f} \right\rceil = \left\lceil \log_2 \frac{100 \times 10^6}{1} \right\rceil = \lceil 26.5754 \rceil = 27 \text{ bits} \qquad (11)$$

here $\lceil\ \rceil$ denotes the ceiling operator. Due to excessive emory requirements, the full precision of the phase accuulator cannot be used to index the sine/cosine look-up ble. A quantized (or truncated) version of the phase angle used for this purpose. The block labeled Q1 in the phase uncation DDS, Figure 1, performs the phase angle quantiation. The lookup table can be located in block or distribed memory.

uantizing the phase accumulator introduces time base jitr in the output waveform. As shown in Eq. (12), this jitter sults in undesired phase modulation that is proportional to e quantization error.

$$\Theta(n) = \theta(n) + \delta\theta(n)$$
$$e^{j\Theta(n)} = e^{j[\theta(n)+\delta\theta(n)]} = e^{j\theta(n)} e^{j\delta\theta(n)}$$
$$e^{j\Theta(n)} \approx e^{j\theta(n)}[1 + j\delta\theta(n)] \qquad (12)$$
$$= e^{j\theta(n)} + j\delta\theta(n) e^{j\theta(n)}$$

Figure 2 shows the look-up table addressing error, complex output time-series and the spectral domain representation of the output waveform produced by the DDS structure shown in Figure 1. The normalized frequency for this signal is 0.022 Hz, which corresponds to phase accumulation steps of 7.92 degrees per output sample. The angular resolution of the 256-point look-up table is 360 / 256 or 1.40625 degrees per address, which is equivalent to 7.92 / 1.40625 or 5.632 addresses per output sample. Since the address must be an integer, the fractional part is discarded and the resultant phase jitter is the cause of the spectral artifacts. Figure 3 provides an exploded view of the spectral plot in Figure 2(c).

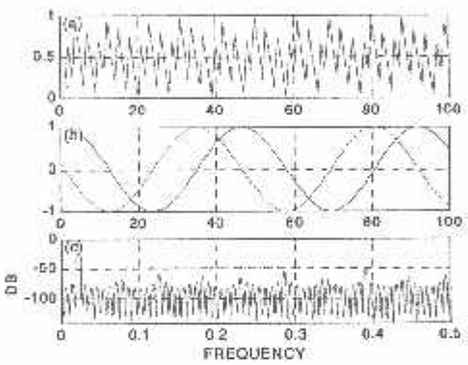


Figure 2: **Phase truncation DDS. $f_{out} = 0.022$ Hz, table depth = 256 12-bit precision samples. (a) Phase angle addressing error; (b) Complex output time series; (c) Output spectrum.**

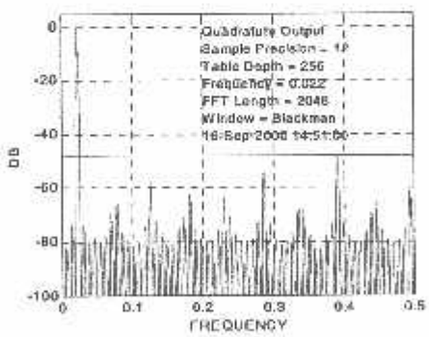


Figure 3: **Phase truncation DDS. $f_{out} = 0.022$ Hz, table depth = 256 12-bit precision samples. Exploded view of Figure 2 (c).**

'e make two observations related to the phase jitter struc-re level. Observe that the fractional part of the address ount is a periodic (sawtooth) error sequence which is sponsible for the harmonic rich (and aliased) low-level ase modulation evident in Figure 3. We also note that the ak distortion level due to incidental phase modulation is proximately 48 dB below the desired signal level, which is nsistent with 6 dB/bit of address space. Put another way, S dB of spur suppression is required in the output wave-rm, as referenced to the 0 dB primary tone, the DDS okup table must support at least $\lceil S/6 \rceil$ address bits. For ample, if $S = 70$ dB, which means that the highest spur ill be 70 dB below the main signal, then the minimum num-er of address bits for the lookup table is $\lceil 70/6 \rceil$ 12 bits; at is, a 4096-deep table.

gure 4 and Figure 5 demonstrate the performance of a milar DDS to the one presented in Figure 2 but in this ample 16-bit precision output samples have been used. bserve that the highest spur is still at the –48 dB level, and locating 4 additional bits to the output samples has not ntributed to any further spur reduction. For a phase trun-ation DDS, the only option to further reduce the spur levels to increase the depth of the look-up table.

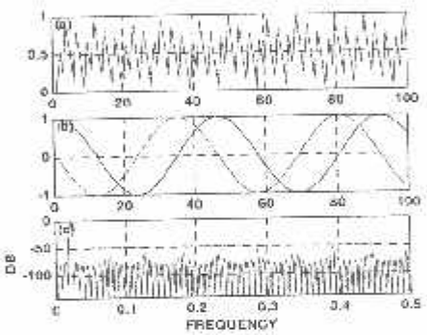


Figure 4: **Phase truncation DDS. $f_{out} = 0.022$ Hz, table lepth = 256 16-bit precision samples. (a) Phase angle addressing error. (b) Complex output time series. (c) Output spectrum.**

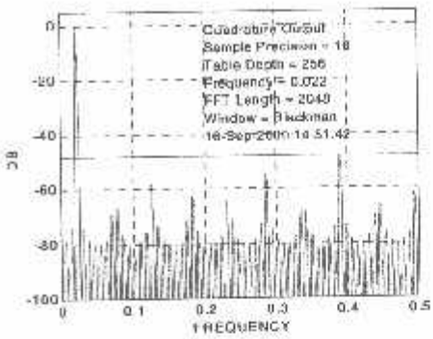


Figure 5: **Phase truncation DDS. $f_{out} = 0.022$ Hz, table depth = 256 16-bit precision samples. Exploded view of Figure 4 (c).**

Further examples illustrating the performance of various DDS configurations are shown in Figure 6 through Figure 15. The configuration details are annotated on the plot. For some of these examples, the synthesized frequency has been swept across a small range of the available output bandwidth. For these cases, the sweep start frequency, stop frequency, frequency increment δf and the number of tones in the sweep interval (*Num Tones*) is indicated. The analysis transform length and window function applied to the output time series is also indicated on the plots.

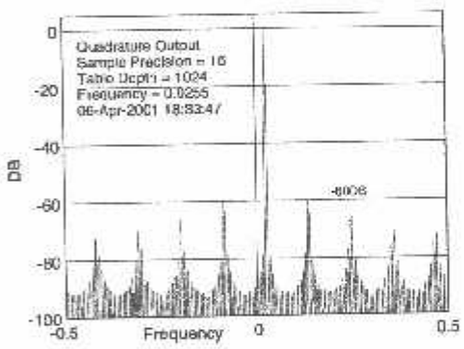


Figure 6:

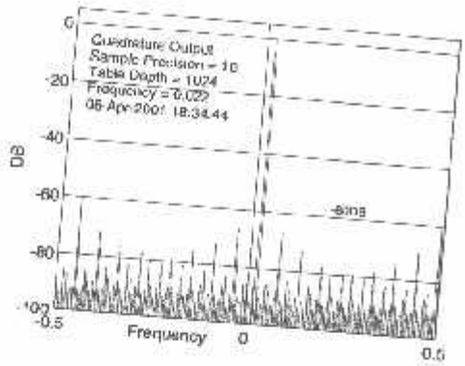


Figure 7:

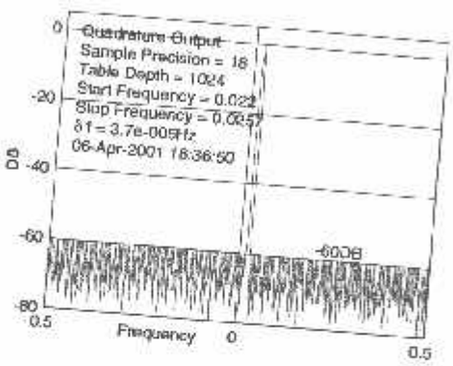


Figure 8:

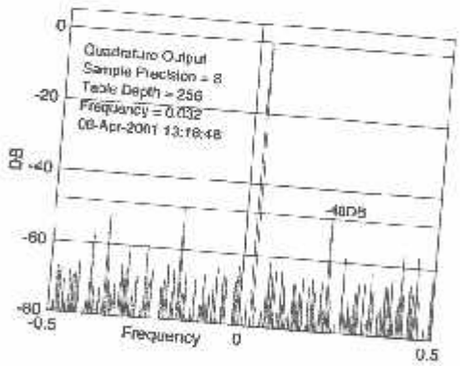


Figure 9:

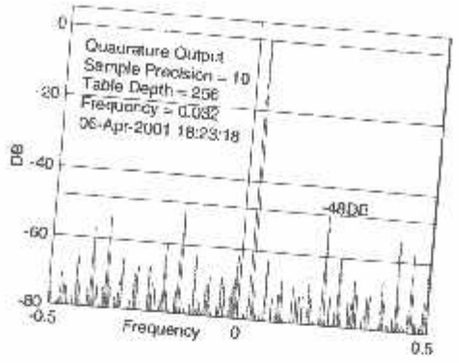


Figure 10:

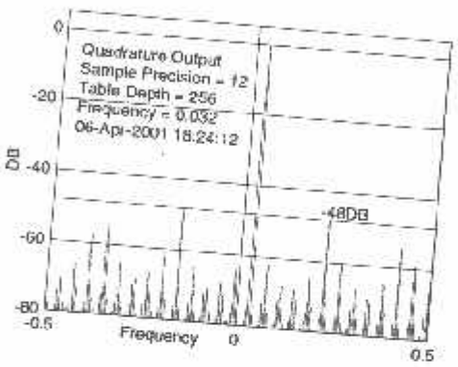


Figure 11:

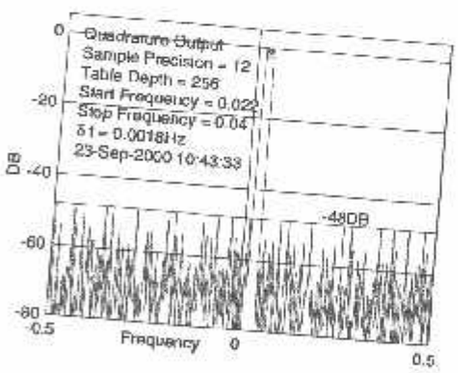


Figure 12:

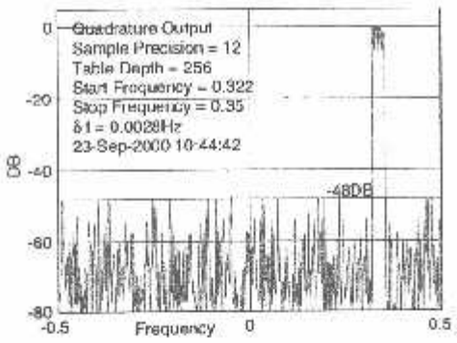


Figure 13:

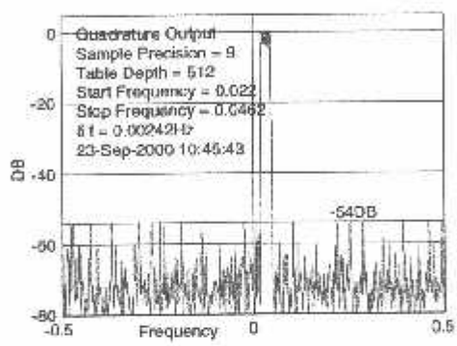


Figure 14:

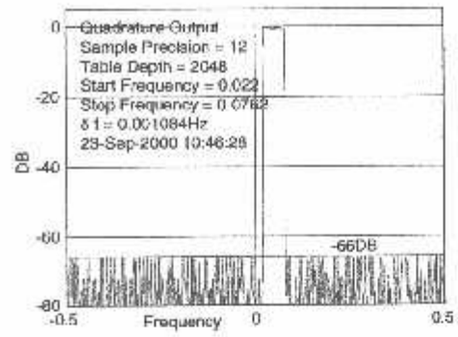


Figure 15:

Figure 1 provides a simplified view of the DDS core. A detailed view is shown in Figure 16. This detailed figure is similar to the simplified view, but also indicates the DDS control and interface *signals CE, A, WE, DATA, RFD* and *RDY*. Also note the inclusion of the *PHASE OFFSET* register designated *POFF*. This register is used for applying a constant phase offset to the phase slope computed in the phase accumulator *PACC*. When the Core is customized, the phase offset source can be defined as either a register, a constant, or it can be omitted entirely. When the *register* option is selected, the phase offset value is supplied via the *DATA* port. The phase offset value is treated as an unsigned quantity. If necessary, the phase offset is zero-extended before it is added to the phase accumulator.

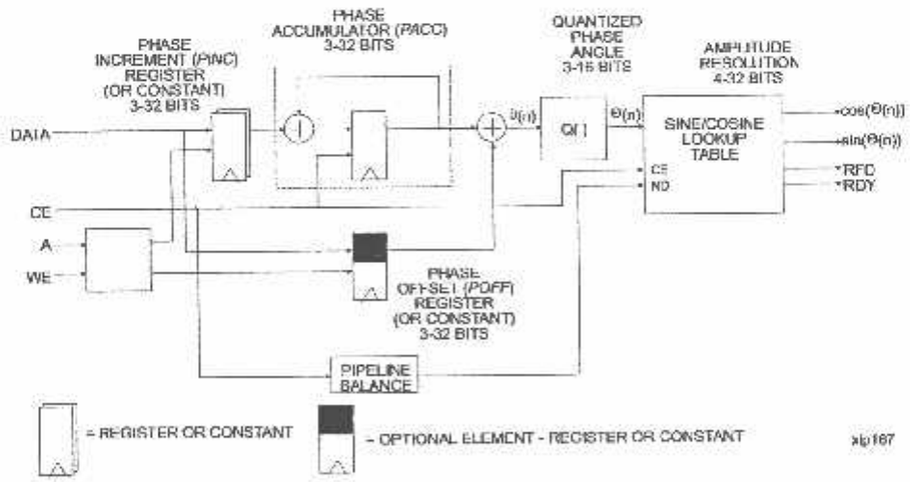


Figure 16: **DDS Core (Detailed View)**

he phase increment value can be sourced from either a register or a constant. When the registered option is selected, the *ATA* port supplies the phase increment value to the phase increment register. When the *constant* option is selected, the DS output frequency is fixed and cannot be adjusted once the core is embedded in a design.

Phase Dithered DDS

n the phase truncation DDS architecture shown in Figure 1 ne quantizer Q1 introduces a phase error in the phase lope by discarding the least significant part, actually frac- onal component, of the high-precision phase accumulator. he phase error due to the discarded fractional part of the ddress count is a periodic series which results in an ndesired spectral line structure. Figure 17 provides an xample of this process for a DDS with a table depth =1024 and table sample precision of 16 bits. Figure 17(a) : the phase error generated by taking the difference etween the quantizer input and output signals, Figure 17) is the output time series and Figure 17(c) is the signal utput spectrum. Observe in Figure 17(a) the periodic saw- ooth structure of the phase error signal. The line spectrum ssociate with this correlated error sequence is impressed n the final output waveform and results in spectral lines in ne synthesizer output spectrum. These spurious compo- ents can be clearly seen in Figure 17(c).

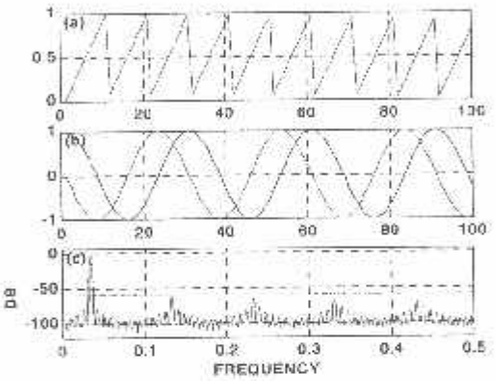


Figure 17: **DDS plots showing (a) phase error time series, (b) complex output time series, (c) output spectrum. 1024 deep lookup table, 16-bit samples, output frequency is 0.333 Hz.**

This structure can be suppressed by breaking up the regularity of the address error with an additive randomizing signal. This randomizing sequence, called *dither*, is a noise sequence, with variance approximately equal to the least significant integer bit of the phase accumulator. The dither sequence is added to the high-precision accumulator output prior to quantization by Q1. The resulting *dithered DDS* architecture is shown in Figure 18.

The dithered DDS supplies, approximately, an additional 12 dB of spurious free dynamic range (SFDR) in comparison to a phase truncation design. The additional logic resources required to implement the dither sequence generator are not significant.

To provide S dB of spur suppression using a phase truncation DDS, as referenced to the 0 dB primary tone, the internal lookup table must support at least $\lceil S/6 \rceil$ address bits. To achieve this same performance using the dithered architecture requires two fewer address bits, minimizing the number of block RAMs (or logic slices for a distributed memory implementation) used in the FPGA implementation. In summary, for a dithered DDS implementation, the number of address bits needed to support S dB spur suppression is equal to $\lceil S/6 \rceil - 2$.

Figure 19, Figure 20, and Figure 21 provide the results for several dithered DDS simulations. Figure 19 shows eight simulations for a complex dithered DDS employing a table depth $N = 4096$ and 16-bit precision samples. For each plot the output frequency is different and is annotated on the plot. A phase truncation design would typically generate output spurs 72 dB below the output frequency, independent of the actual value of the output frequency. Indicated on each of the plots by the parameter A is the peak spur level achieved for the simulation. The eight spurs are -88.12, -88.22, -86.09, -88.80, -87.21, -87.55, -87.83, -87.12 dB below the output frequency. The worst case value of -86.09 is 14.09 dB better than a similarly configured phase truncation DDS.

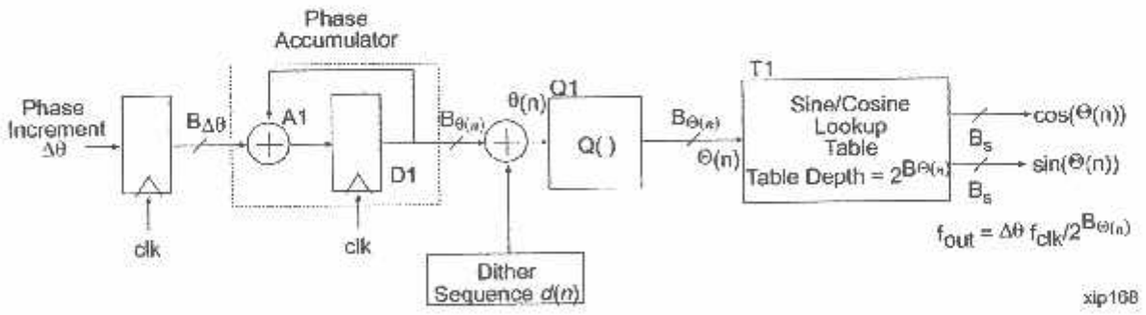


Figure 18: **Phase Dithered DDS Architecture**

achieve this same SFDR by extending the table length of phase truncation design would require extending the table more than a factor of four.

gure 20 and Figure 21 provide two more dithered DDS nulations where the output frequency is swept over a band of frequencies. The spectrum for each discrete tone in the sweep band is overlaid to construct the final plot. The sweep start frequency, end frequency, number of tones in the sweep, and DDS configuration are annotated on the plot.

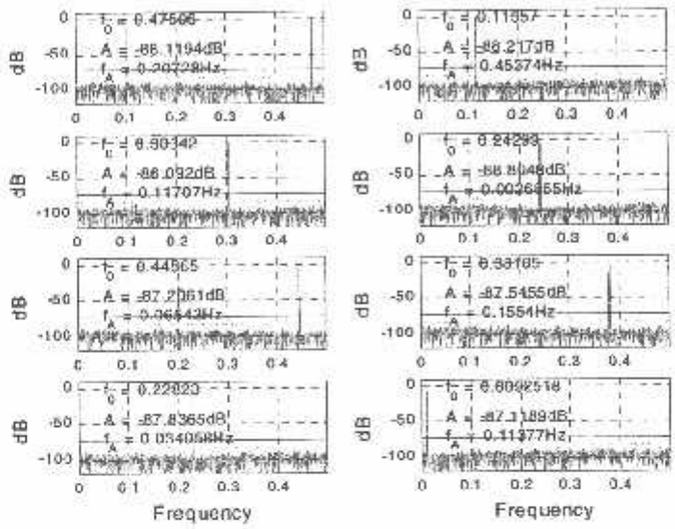


Figure 19: **Dithered DDS Simulations. The DDS configuration is $N = 4096$, $B_r = 16$. The eight plots are spectral domain representations for eight different output frequencies. Each plot is annotated with the peak spur level.**

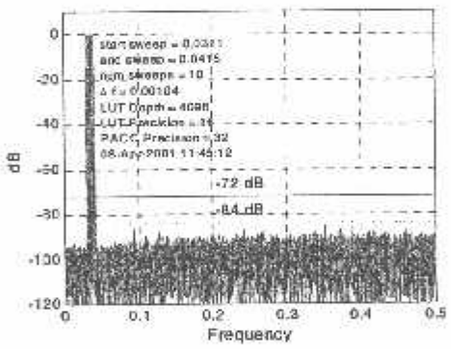


Figure 20:

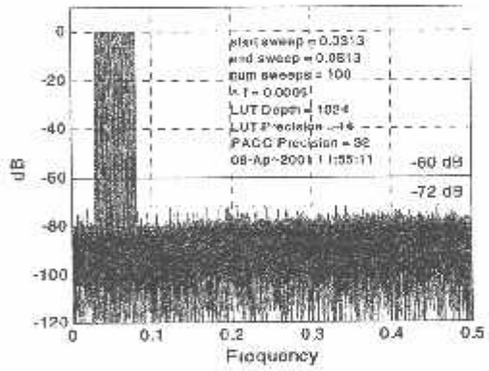


Figure 21:

Figure 20 the synthesized signal is swept over a range of equencies starting from 0.0311 to 0.0415 Hz. There are n tones in the sweep separated in frequency by 0.00104 z. In this example the phase truncation DDS would pro- ice peak spurs at –72 dB with respect to the 0 dB primary gnal. The dithered DDS provides approximately 12 dB better performance with the peak spur –84 dB below the output signal.

Figure 21 shows the results of a second swept frequency dithered DDS simulation. In this case the start frequency is 0.0313 Hz, the sweep termination frequency is 0.0813 Hz and there are 100 tones in the sweep. The frequency differential between successive simulations is 0.0005 Hz. A sim-

arly configured phase truncation DDS, with a table depth of 024 16-bit samples, would produce spurs approximately) dB below the output signal. Observe from the plot that ıe dithered DDS generates spurs that are 72 dB below the dB output signal.

further advantage of the dithered DDS is that the spectral ıe structure present in a phase truncation design is ımoved and the out-of-band signal is significantly whit- ıed. This white broadband noise floor is more desirable ıan the line structured spectrum. In digital communication ceivers that use a DDS for generating mixing signals for erforming channelization functions, the spurs in a phase uncation DDS can act as low-level mixing tones and cause ıdesirable spectral contamination of the desired channel. or virtually all applications the preferred implementation is ıe dithered DDS.

aylor Series Corrected DDS

ıe phase dithered DDS, as well as the phase truncation DS, have a quantizer Q1 which produces a lower precision ;n), by discarding the fractional component of the high pre- sion e(n). The reason for this quantization step is to keep e size of the lookup memory to a reasonable size. The ade-off is spectral purity. With the availability of embedded ultipliers in the Virtex-II, Virtex-II Pro, Spartan-IIE, and partan-3 FPGAs, it is now practical, from a silicon-area andpoint, to use the previously discarded fractional bits to alculate corrections that can be added to the lookup table alues and produce outputs with very high spurious free ynamic range (SFDR). These embedded multipliers are SIC like multipliers and do not consume any of the logic bric. The Taylor series correction DDS fixes a number of e parameters as described in the Parameters section nder Noise Shaping. The additional resources required ver the phase truncation DDS are two embedded multipli- s, one constant coefficient multiplier, and four adders.

gure 22, Figure 23, Figure 24, and Figure 25 show the sults of four different Taylor series corrected DDS simula- ons. The Taylor series corrected architecture uses a table epth $N = 4096$ and 18-bit precision samples. However, the ecision at the output of the feed-forward error processor is) bits. For each plot the output frequency is different and notated directly on the plot. A similarly configured phase uncation DDS would produce spurs at –72 dB and a phase thered DDS at –84 dB. The peak spurs for the four plots e –118.25, –118.13, –118.10, and –118.17 dB below the utput frequency.

gure 27 shows a swept frequency Taylor series corrected DS. The starting frequency for this example is 0.0313 Hz, e final frequency is 0.0813 and there are 100 tones in the veep. Using this configuration, a phase truncation DDS ould produce peak spurs at approximately 72 dB below ıe output signal and a phase dithered DDS would produce eak spurs at approximately 84 dB below the output signal. s shown in the plot the Taylor series corrected DDS pro- duced spurs that were all the way down to 118 dB below the output signal. This result is 34 dB better than the phase dith- ering DDS, 46 dB better than the phase truncation DDS, and still only consumes a single 18Kb block RAM for the lookup storage. Figure 28 shows another frequency sweep simulation with 35 tones over a broader frequency range.

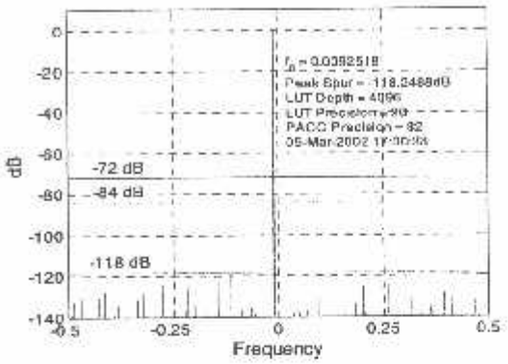


Figure 22: **Taylor series corrected DDS – single tone test. $f_0 = 0.0092518$.**

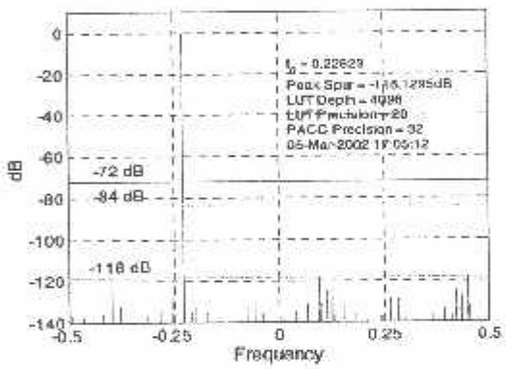


Figure 23: **Taylor series corrected DDS - single tone test. $f_0 = 0.22823$.**

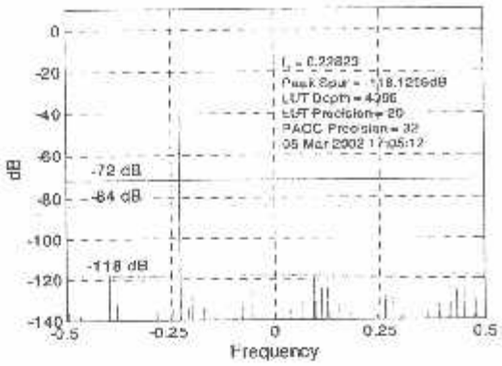


Figure 24: **Taylor series corrected DDS - single tone test. $f_0 = 0.22823$.**

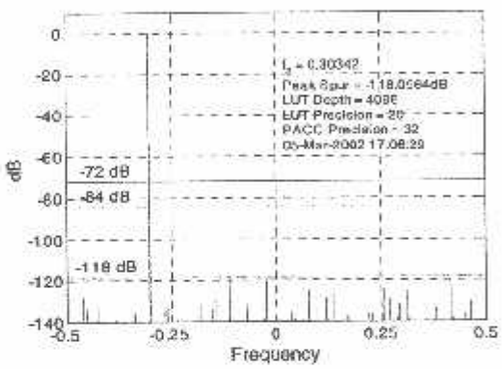


Figure 25: **Taylor series corrected DDS - single-tone test. $f_0 = 0.30342$.**

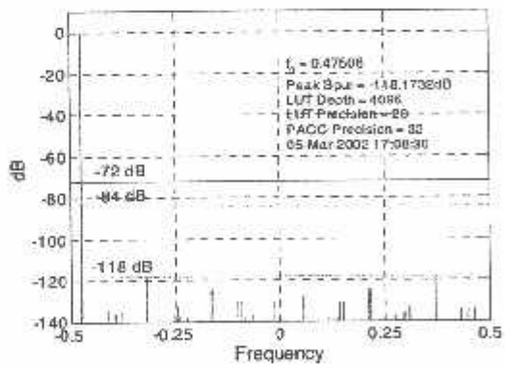


Figure 26: **Taylor series corrected DDS - single tone test. $f_0 = 47506$.**

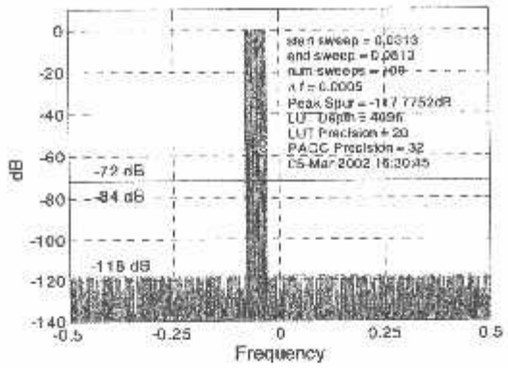


Figure 27: **Taylor series corrected DDS - frequency sweep simulation. 100 tones.**

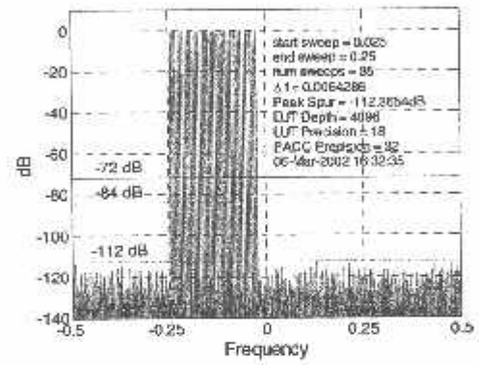


Figure 28: **Taylor series corrected DDS - frequency sweep simulation. 35 tones.**

Interface, Control, and Timing

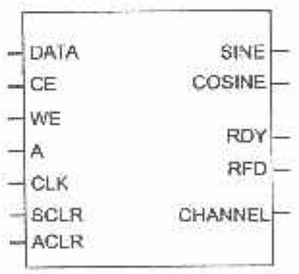


Figure 29: **DDS Symbol**

able 1: **Core Signal Pinout**

Signal Name	Direction	Description
CLK	Input	Master Clock - active rising edge.
A†	Input	Address select for writing to the phase increment (*PINC*) memory and the phase offset (*POFF*) memory. When A_{MSB}=0, the *PINC* memory is selected. When A_{MSB}=1, the *POFF* memory is selected. The four lower order bits of A are used to address up to 16 channels for the currently selected memory. The memory map is shown in Table 2.
WE†	Input	Write enable - active high. Enables a write operation to the *PINC* or *POFF* memories.
CE†	Input	Clock enable - active high. *CE* must be high during normal Core operation, but it is not required to be active during a write access to the *PINC* or *POFF* memories.
DATA†	Input	Time shared data bus. The *DATA* port is used for supplying values to the *PINC* or *POFF* memories.
ACLR†	Input	Asynchronous clear - active high. When *ACLR* is asserted, the all registers in the Core are cleared. *RDY* is also deasserted.
SCLR†	Input	Synchronous clear - active high. When *SCLR* is asserted, all registers in the Core are cleared. *RDY* is also deasserted.
RDY†	Output	Output data ready - active high. Indicates when the output samples are available.

Table 1: **Core Signal Pinout *(Continued)***

Signal Name	Direction	Description
RFD†	Output	Ready for data - active high. *RFD* is a dataflow control signal present on many Xilinx LogiCOREs. In the context of the DDS, it is supplied only for consistency with other LogiCOREs. This optional port is always tied to VCC.
CHANNEL†	Output	Channel index. Indicates which channel is currently available at the output when the DDS is configured for multi-channel operation. This is an unsigned two's complement signal. It's width is determined by the number of channels.
SINE	Output	Sine time-series.
COSINE	Output	Cosine time-series.

† denotes optional pin

Figure 30 shows the timing sequence for a single-channel DDS core. In this example, the DDS has both a phase increment *(PINC)* and a phase offset *(POFF)* memory. The *PINC* memory is first written with the *PINC* value 0x0010. This is realized by supplying the value on the *DATA* port and addressing the *PINC* memory by defining the MSB of the *A* port as A_{MSB}=0, as shown by the memory map in Table 2. Since this example is a single-channel case, the remaining four bits of the *A* port should be set to zeros. The write is performed on the positive clock edge. *WE* must be active; *i.e.*, *WE*=1, to perform this operation. Immediately after the *PINC* memory is loaded, the value 0x1000 is written to the *POFF* memory. This requires A_{MSB}=1 and *WE*=1.

Table 2: **Phase Increment and Phase Offset memory map**

Address	Location Description
00000 - 01111	Phase Increment (PINC) values for channel 0 through 15
10000 - 11111	Phase Offset (POFF) values for channel 0 through 15

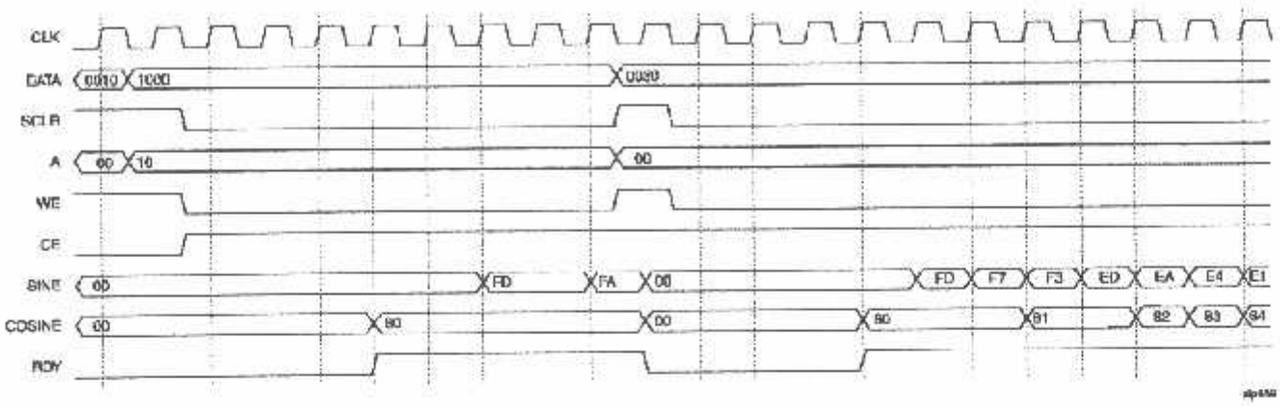


Figure 30: **DDS Timing: single-channel**

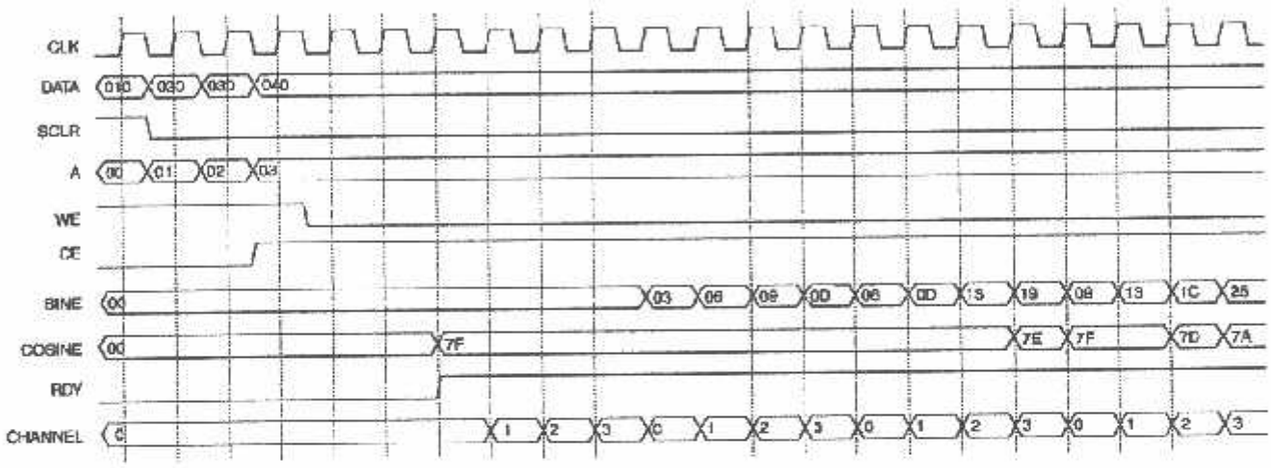


Figure 31: **DDS Timing: multichannel**

ıe memory is loaded on the rising edge of the clock. *CE* ›es not have to be active to write to either the *PINC* or *OFF* memories. The DDS will start operating once the ›ck enable is applied (*CE*=1). Since *CE* is an optional pin, DS configurations that do not include this pin will begin ıerating once the FPGA is configured and the system ›ck is active. After a start-up latency (measured from the ısertion of *CE*[2]) that depends on the pipelining configura-›n chosen for the core samples will be presented on the ıtput port(s). This is indicated by *RDY*=1. For most config-ations, the assertion by the Core of *RDY* indicates the first ılid output sample. However, there is an exception. If a DS is customized such that the *0-cycle* latency phase :cumulator option is selected, and the sine-cosine look-up ble is in distributed memory, and the table is purely combinatorial, any writes to the *PINC* register will be immediately reflected at the output port(s). This is irrespective of whether *CE* is asserted or not. In this situation, there are no registers between the *PINC* register and the output nodes, there is only a combinatorial arrangement of logic. Therefore, the *CE* pin cannot have any influence on this path through the system. The *CE* pin will, of course, still control the operation of the register (now in the upper arm of the phase accumulator shown in Figure 36 (b)) in the *PACC*. As illustrated in Figure 30, valid samples begin appearing at the output ports when *RDY* goes high.

The DDS can have an optional asynchronous clear or synchronous clear port. When either type of clear is applied, the *SINE* and *COSINE* output ports will assume a value of

Assuming this part is present.

Figure 30 shows the *SCLR* being asserted and a new *'NC* value being written on the same clock. This results in DY going low until the pipeline fills, and then back high hen valid samples are available again. The new *PINC* alue is 0x0030, which represents an output frequency that three times that of the previous *PINC* value. The new out- it samples can be seen to be changing at a faster rate, as pected.

gure 31 shows the timing for a four-channel DDS imple- entation. The *PINC* value for each channel is written on e first four clock cycles. Valid samples are available on the itputs when the *RDY* signal goes high. The additional *HANNEL* port indicates which channel is currently avail- ile at the output.

arameters

ie DDS parameterization screens are shown in Figure 32, gure 33, Figure 34, Figure 36, and Figure 37.

ie customization parameter definitions are:

Component Name: The user-defined DDS component name.

Function: The DDS may have a quadrature output (sine and cosine), or a single output port — either sine or cosine. In addition, the sign of the output signal(s) can be defined using the *Negative Sine* and *Negative Cosine* checkboxes. For example, if the quadrature output option is selected (*Sine and Cosine* GUI option), and both the *Negative Sine* and *Negative Cosine* boxes are unchecked, then the output signal *s(n)* is

$$s(n) = e^{j\Theta(n)} = \cos\Theta(n) + j\sin\Theta(n) \quad (13)$$

If the Negative Sine box has been checked, the output signal is defined by

$$s(n) = e^{-j\Theta(n)} = \cos\Theta(n) - j\sin\Theta(n) \quad (14)$$

- Channels: The DDS core can generate a single-channel implementation as well as a multichannel implementation with support for up to 16 independent channels. If a multichannel implementation is generated, all channels will time-share the DDS outputs.
- DDS Performance Options: Both system-level and circuit-level performance requirements are specified and the DDS core generates an implementation to meet these requirements
  - **DDS Clock Rate:** The frequency at which the DDS core will be clocked
  - **Spurious Free Dynamic Range (SFDR):** This parameter defines the frequency domain requirements of the out-of-band noise generated by the DDS outputs. The range is from 16 to 115 dB of spur suppression. Note that an SFDR value of 102 dB or greater will force an implementation employing a Taylor Series Correction which requires the use of embedded multipliers.
  - **Frequency Resolution:** This parameter determines the granularity of the tuning frequency. If the value entered is 10, the tuning frequency can be adjusted to a precision of 10 Hz. As an example, you could tune the DDS to a frequency of 5.00003 MHz.

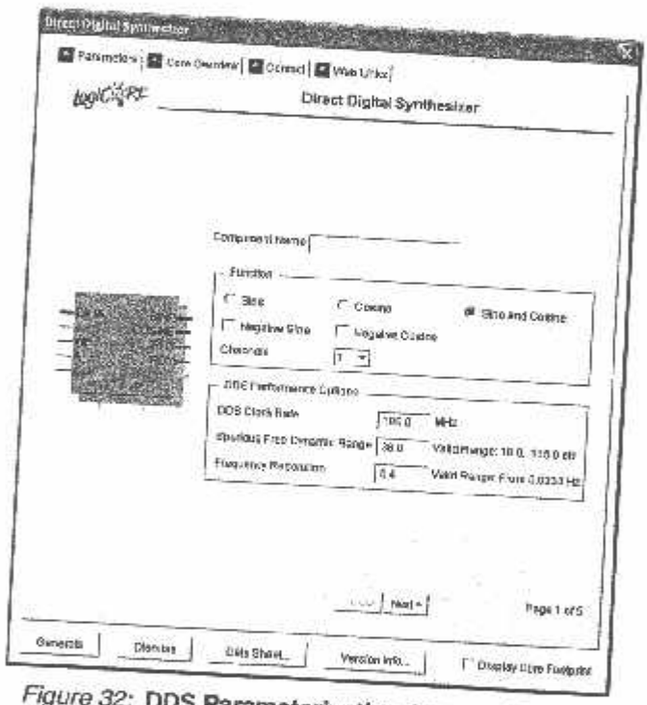


Figure 32: **DDS Parameterization Screen - Field 1**

Output Frequencies: For each channel, an independent frequency can be entered into the table for the Sine and Cosine outputs. The allowable upper range is displayed above the table and is calculated by taking one half of the DDS clock rate and then dividing by the total number of channels. This upper limit is required so that the DDS clock rate does not drop below the Nyquist frequency.

- **Phase Increment:** The user has the choice of making the output frequency a constant value or in-circuit changeable. Selecting *Fixed* will make the frequencies constant and selecting *Programmable* will allow the phase increment values to be changed in-circuit. If *Programmable* is selected, the values entered in the table will be the initial frequencies that take effect once the FPGA has been configured. If an ACLR or SCLR signal is asserted, the output frequencies settings will not be altered. Note that the *Fixed* or *Programmable* option is applied for all DDS channels.

Phase Offset Angles: An independent offset can be added to the phase angle of each channel by entering a value into the table. The entered values will be multiplied by 2π radians. The valid range is -1.0 to 1.0.

- **Phase Offset:** The user has the choice of making the phase offset angles a constant value, in-circuit changeable, or not used. Selecting *Fixed* will make the offsets constant; selecting *Programmable* will allow them to be changed in-circuit; selecting *None* will prevent any offset from being added. If *Programmable* is selected, the values entered in the table will be the initial offsets that will take effect once the FPGA has been configured. If an ACLR or SCLR signal is asserted, the phase offset angles settings will not be altered. Note that the *Fixed*, *Programmable*, or *None* option is applied for all DDS channels.

- **Clear Options:** If the *ACLR Pin* is selected, the core will be generated with a asynchronous reset. If the *SCLR Pin* is selected, the core will be generated with a reset that is synchronized to the clock. When asserted, the internal logic returns to its initialized state. Note that all programmable values are retained. The Sine and Cosine output ports will be driven to zeros until enough clock cycles have passed to fill the core's internal pipeline, which can be determined by the Latency value.
- **Clock Enable:** The Core can have an optional clock enable port.

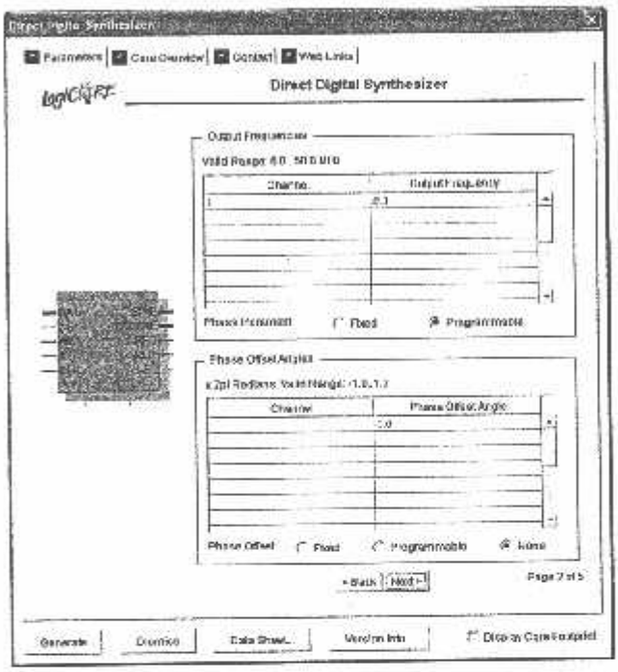


Figure 33: **DDS Parameterization Screen - Field 2**

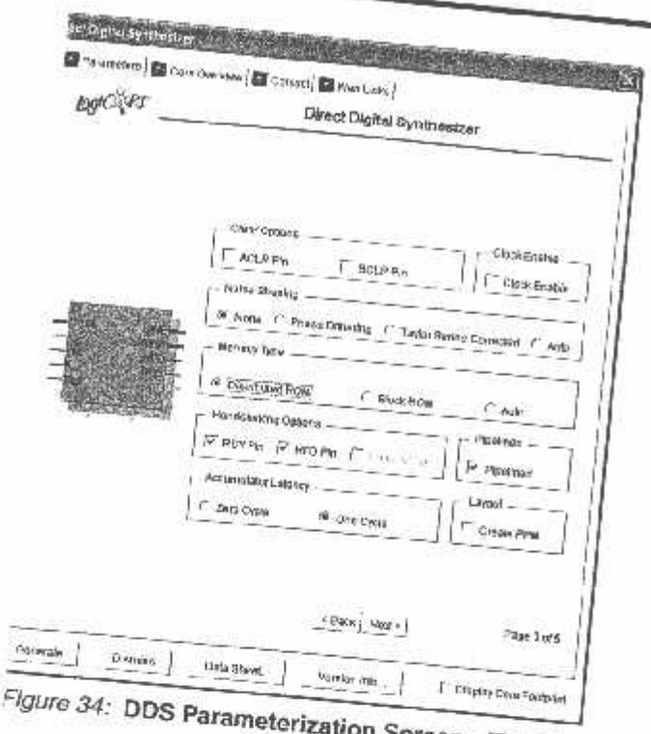


Figure 34: **DDS Parameterization Screen - Field 3**

Noise Shaping: The radio buttons in this frame control whether a phase truncation, dithered DDS, or Taylor series corrected DDS implementation are generated. When *Auto* is selected, the noise-shaping type will be automatically determined, based on other core parameters including *SFDR*. When *None* is selected, a phase truncation DDS is produced. When *Phase Dithering* is selected, a dithered DDS implementation is generated. When *Taylor Series Corrected* is selected, a Taylor series corrected DDS implementation is generated. Taylor series corrected implementations are supported for only Virtex-II, Virtex-II Pro, Spartan-IIE, and Spartan-3 FPGAs. The *Taylor Series Corrected* radio button will be disabled when other architectures are selected. When *Taylor Series Corrected* is selected, the accumulator width will be fixed to 32 bits; the phase angle width will be fixed to 12 bits; the memory type will be fixed to block ROM; outputs required will be fixed to sine and cosine; negative sine will be fixed to true; negative cosine will be fixed to false; pipelining will be fixed to true, and the output width will be fixed to 20 bits.

- **Memory Type:** This field controls the location of the DDS trigonometric lookup table. When *Distributed ROM* is selected, the table is placed in distributed memory. If *Block ROM* is selected, the table will be implemented using block memory. If *Auto* is selected, the actual memory type will be automatically determined, based on other core parameters, including *SFDR*.
- **Handshaking Options:** Optional handshaking ports—*RDY* and *RFD*—can be included on the Core. The *RFD* output signal is simply tied to VCC and is an optional port that can be included for compatibility with other Xilinx LogiCORE products that employ this style of dataflow interface. As shown in Figure 30, the *RDY* output signal identifies when valid sine/cosine samples appear on these ports after the Core is started from rest — either after system power-on or a reset (synchronous or asynchronous). Any type of core reset will cause RDY to be removed (*RDY*=0). When the *Channels* parameter is set to a value of two or more and the *Channel Pin* parameter is selected, an additional output will be generated to indicate which channel the current output samples belong to.
- **Pipelined:** When *Pipelined* is selected, the core will be generated with pipelining registers inserted throughout the datapath. The insertion of pipeline registers enables the core to run at higher clock rates by shortening the delays between register stages. Pipelining increases the latency of the core, which is reported by the *Latency* value on the summary page of the GUI (Figure 35). If *Pipelined* is not selected, the latency of the core will be zero when a distributed memory implementation has been selected and the latency will be one when a block memory implementation has been selected.

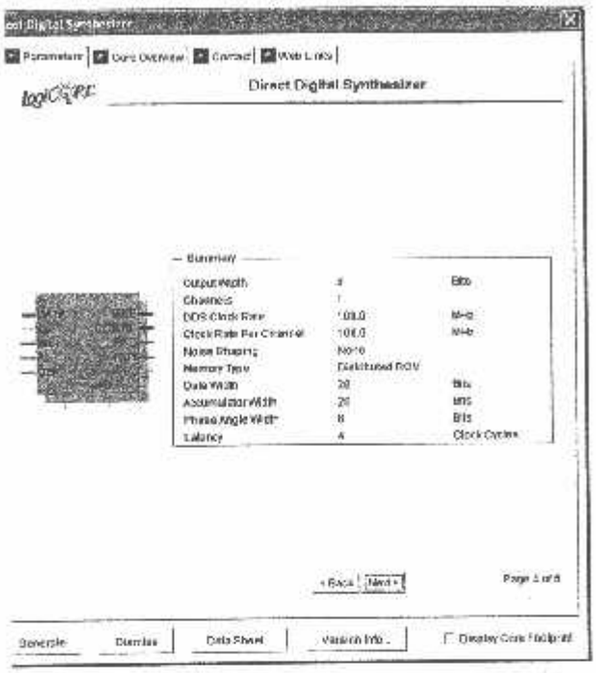


Figure 35: **DDS Parameterization Screen - Field 4**

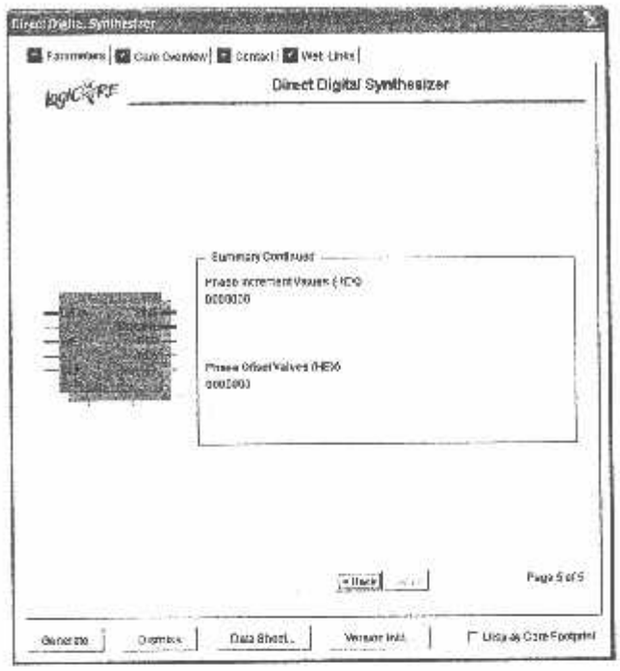


Figure 37: **DDS Parameterization Screen - Field 5**

Phase Accumulator: Phase accumulator precision. This field defines the precision of the *PACC* register (Figure 16). The location of the register in the phase accumulator is controlled by the latency selection options. When the *one-cycle* latency option is selected, the phase accumulator will be as shown in Figure 36(a). When the *zero-cycle* option is selected, the arrangement in Figure 36(b) is employed.

Layout: This checkbox controls whether a relationally placed MACRO (RPM) or a module with no placement information is generated. When checked, an RPM is produced.

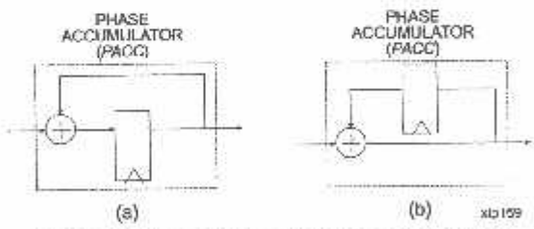


Figure 36: **Register Options for the Phase Accumulator--PACC**

Table 3: **Virtex Core Resource Utilization for Various Table Sizes (*Distributed ROM Implementation: 32-bit Data Width; 32-bit Phase Accumulator Width*)**

Phase Angle Width	Output Width	Single Output	Sine and Cosine
6*	8	32	49
	12	40	65
	16	48	81
	32	80	145
8	8	43	71
	12	53	91
	16	63	111
	32	103	191
10	8	95	175
	12	133	251
	16	171	327
	32	323	631

* Fullwave gets stored.

Note: Slice count is an approximation.

able 4: Virtex-II Core Resource Utilization for Various Table Sizes (*Distributed ROM mplementation: 32-bit Data Width; 32-bit Phase Accumulator Width*)

Phase Angle Width	Output Width	Single Output	Sine and Cosine
6*	8	32	49
	12	40	65
	16	48	81
	32	80	145
8*	8	80	145
	12	112	209
	16	144	273
	32	272	529
10	8	89	162
	12	123	230
	16	157	296
	32	292	568

Fullwave gets stored.
ote: *Slice count is an approximation.*

able 5: Virtex Core Resource Utilization for Various Table Sizes (*Block ROM Implementation: 32-bit Data Width; 2-bit Phase Accumulator Width*)

Theta Width	Output Width	Single Output		Sine and Cosine	
		No. of Slices	No. of Block ROMs	No. of Slices	No. of Block ROMs
6	8	16	1*	17	1*
	12	16	1*	17	1*
	16	16	1*	17	1*
	32	16	1*	62	2
8	8	16	1*	17	1*
	12	16	1*	17	1*
	16	16	1*	17	1*
	32	42	1	68	2
10	8	33	1	50	1
	12	35	1	54	1
	16	37	1	58	1
	32	45	2	74	2
13	8	37	4	59	4
	12	39	6	63	6
	16	41	8	67	8
	32	48	16	81	16

ble 5: **Virtex Core Resource Utilization for Various Table Sizes (*Block ROM Implementation: 32-bit Data Width; -bit Phase Accumulator Width*) *(Continued)***

Theta Width	Output Width	Single Output		Sine and Cosine	
		No. of Slices	No. of Block ROMs	No. of Slices	No. of Block ROMs
16	8	69	28	120	28
	12	87	44	156	44
	16	105	60	192	60
	32	239	124	461	124

ble 6: **Virtex-II Core Resource Utilization for Various Table Sizes (*Block ROM Implementation: 32-bit Data Width; -bit Phase Accumulator Width*)**

Phase Angle Width	Output Width	Single Output		Sine and Cosine	
		No. of Slices	No. of Block ROMs	No. of Slices	No. of Block ROMs
6	8	16	1*	17	1*
	12	16	1*	17	1*
	16	16	1*	17	1*
	32	16	1*	17	1*
8	8	16	1*	17	1*
	12	16	1*	17	1*
	16	16	1*	17	1*
	32	16	1*	17	1*
10	8	16	1*	17	1*
	12	16	1*	17	1*
	16	16	1*	17	1*
	32	45	1	74	1
13	8	36	1	57	1
	12	38	2	61	2
	16	40	2	65	2
	32	49	4	83	4
16	8	39	7	62	7
	12	63	10	110	10
	16	65	14	114	14
	32	122	28	226	28

ulwave gets stored. *Note: Slice count is an approximation.*

ıe additional logic required for the phase dithered DDS is 55 slices. The total slice count can be calculated by adding 55 any of the slice count data shown in Table 3, Table 4, Table 5, and Table 6.

ıe utilization for the Taylor series corrected DDS is as follows: 315 slices; one 18Kb block RAM, and two 18x18s embed-ɔd multipliers.

CO File Parameters

ıe parameters supplied via the filter GUI are captured and logged to the.*xco* file. The full name of this file is simply the ɔmponent Name with an .xco file extension. Table 7 defines the .xco file parameter names and range specifications. gure 38 is an example .xco file.

ble 7: **XCO File Parameter Names, Definitions, and Range Specifications**

Parameter Name	Definition	Range
ɔusFormat	Controls the notation employed for identifying buses in the output edif netlist file	{BusFormatAngleBracket | BusFormatParen}
ɔimulationOutputProducts	Core HDL simulation selection — either VHDL or Verilog	{VHDL | VERILOG}
'iewlogicLibraryAlias	Pathname to Viewlogic directory	Valid path name for the user's operating system
ɔilinxFamily	The FPGA target device family	{Virtex | Virtexe | Virtex2 | Virtex2p | Spartan2 | Spartan2e | Spartan3}
ɔesignFlow	HDL flow specifier	{VHDL | VERILOG}
ɔowVendor	Design flow vendor information	{Other | Synplicity | Exemplar | Synopsis | Foundation}
hannels	The number of independent channels the generated core will support	[1,...,16]
reate_rpm	When *create_rpm=true* a Core with embedded physical placement information is generated. If *create_rpm=false* the Core is generated without placement data.	{true | false}
ɔelined	This parameter controls the degree of pipelining employed in the sine/cosine lookup table. When set to *true* the table is fully pipelined. When set to *false* the minimum (zero for the case of a distributed ROM) number of pipeline registers are used.	{true | false}
oise_shaping	When *noise_shaping=Taylor_Series_Corrected*, a Taylor series corrected DDS is generated. When *noise_shaping=Phase_Dithering*, a dithered DDS is generated. When *noise_shaping=None*, a phase truncation DDS is generated. When *noise_shaping=Auto*, the noise shaping type will be determined by the core.	{Taylor_Series_Corrected | {Phase_Dithering | None | Auto}
lock_enable	Clock enable. When *ce_enable=true* a clock enable port is included on the component. When *clock_enable=false* the clock enable port suppressed.	{true | false}
ɔurious_free_dynamic_range	A quantity that describes the spectral purity of the output waveforms	[18,...,115]

ble 7: **XCO File Parameter Names, Definitions, and Range Specifications** ***(Continued)***

Parameter Name	Definition	Range
ccumulator_latency	The user may control the position of the register in the phase accumulator. When *accumulator_register=ONE_CYCLE* the register location shown in Figure 38(a) is employed. When *accumulator_register=ZERO_CYCLE*, the register location shown in Figure 38(b) is employed.	{ONE_CYCLE I ZERO_CYCLE}
clr_pin	Synchronous clear. When *sclr_pin=true*, a synchronous reset port is included on the component. When *sclr_pin=false*, the synchronous re reset port suppressed.	{true I false}
dy_pin	When *rdy_pin=true*, the DDS Core includes a *RDY* output signal. If *rdy_pin=false*, the *RDY* port is suppressed.	{true I false}
nemory_type	The sine/cosine samples can be stored in distributed or block memory. When *memory_type=Block_ROM*, the samples are stored in Block memory. When *memory_type=Distributed_ROM*, distributed memory is used. When *memory_type=Auto*, the memory type will be determined by the core.	{Block_ROM I Distributed_ROM I Auto}
hase_Increment	The phase increment value, that is the delta phase increment supplied to the phase accumulator, may be sourced (by the *PACC*) from either a memory (*phase_increment=Programmable*) or a constant (*phase_increment=Fixed*).	{Programmable I Fixed}
hannel_pin	If *channels* is greater than one, a *CHANNEL* port can optionally be added	{true I false}
ds_clock_rate	Frequency of the DDS clock signal	[0,...,300]
clr_pin	Asynchronous clear. When *aclr_pin=true*, an asynchronous reset port is included on the component. When *aclr_pin=false*, the asynchronous reset port suppressed.	{true I false}
omponent_name	Textbox that defines the DDS component name	Any valid file name for the user's operating system consisting of the letters a...z, 0...9 and '_'. The component name may be a maximum of 32 characters.
egative_sine	Applicable only if a sine port has been specified for the particular core instance. When *negative_sine=false*, the signal presented at the *sine* port is sin $\theta(n)$. When negative_sine=true, the signal presented at the sine port is - sin $\theta(n)$.	{true I false}

Table 7: XCO File Parameter Names, Definitions, and Range Specifications *(Continued)*

Parameter Name	Definition	Range
outputs_required	The DDS may be customized to provide a sine-only output (*function=Sine*), cosine-only output (*function=Cosine*) or both sine and cosine (quadrature) outputs (*function=Sine_and_Cosine*).	{Sine | Cosine | Sine_and_Cosine}
utput_frequencies	A comma-delimited set of frequency values. The number of values is determined by the *channels* value. Frequency values are decimals.	
egative_cosine	Applicable only if a cosine port has been specified for the particular core instance. When *negative_cosine=false*, the signal presented at the *cosine* port is cos $\theta(n)$. When *negative_cosine=true*, the signal presented at the cosine port is - cos $\theta(n)$.	{true | false}
d_pin	When *rfd_pin=true*, the DDS core includes a *RFD* output signal. If *rfd_pin=false*, the *RFD* port is suppressed.	{true | false}
hase_offset_angles	A comma-delimited set of phase offset angle factors. The number of values is determined by the *channels* value. Phase offset values are decimals from -1.0 to 1.0.	
equency_resolution	A decimal value which determines the granularity of the *output_frequencies*	
hase_offset	As shown in Figure 16, an optional phase offset can be introduced at the output of the phase accumulator. When *phase_offset=Programmable*, this value is supplied by a memory. When *phase_offset=Fixed*, the phase offset is a constant that is supplied by the *phase_offset_angles* field when the core is elaborated. If *phase_offset=None*, no phase offset is included in the datapath.	{Programmable | Fixed | None}

```
# Xilinx CORE Generator 5.1.02i; Cores Update # 1
# Username = nyquist
# COREGenPath = c:\xilinx\coregen
# ProjectPath=C:\xilinx_projects\DSPGroup\ip_portfolio\dds\core
# ExpandedProjectPath = C:\xilinx_projects\DSPGroup\ip_portfolio\dds\core
# Core name: xco_example
SET BusFormat = BusFormatParen
SET SimulationOutputProducts = VHDL
SET ViewlogicLibraryAlias = ""
SET XilinxFamily = Virtex2
SET DesignFlow = VHDL
SET FlowVendor = Synplicity
SELECT Direct_Digital_Synthesizer Virtex2 Xilinx,_Inc. 4.2
CSET channels = 1
CSET create_rpm = true
CSET pipelined = true
CSET noise_shaping = Taylor_Series_Corrected
CSET clock_enable = true
CSET spurious_free_dynamic_range = 104
CSET accumulator_latency = One_Cycle
CSET sclr_pin = false
CSET rdy_pin = true
CSET memory_type = Block_ROM
CSET phase_increment = Programmable
CSET channel_pin = false
CSET dds_clock_rate = 100.0
CSET aclr_pin = false
CSET component_name = xco_example
CSET negative_sine = true
CSET outputs_required = Sine_and_Cosine
CSET output_frequencies = 33.333
CSET negative_cosine = false
CSET rfd_pin = false
CSET phase_offset_angles = 0.0
CSET frequency_resolution = 0.4
CSET phase_offset = None
GENERATE
```

Figure 38: **Example DDS XCO File**

)S Design Example 1

sider a DDS that is to satisfy the following requirements

frequency resolution $\Delta f = 0.25$ Hz

minimum spur suppression $S = 96$ dB

$f_{clk} = 120$ MHz

Quarature output: $s(n) = e^{-j(\Theta(n)+\phi)} = \cos(\Theta(n)+\phi) - j\sin(\Theta(n)+\phi))$

where the phase offset $\phi = \pi/4$ radians

DDS GUI accepts system-level parameters instead of evel parameters such as the width of the phase accutor, width of the phase angle, etc. Because of this all of equirements above can be entered into the GUI directly ut having to calculate low-level core details. To illus- this example 1 will show some calculations that the does automatically.

field width of the phase accumulator (PACC in e 29) defines the DDS frequency resolution according

$$\Delta f = \frac{f_{clk}}{2^{B_{PACC}}} \tag{15}$$

Therefore, the bit-precision for the PACC register is

$$\begin{aligned} B_{PACC} &= \log_2\left\lceil \frac{f_{clk}}{\Delta f} \right\rceil \\ &= \left\lceil \log_2 \frac{120\times 10^6}{0.25} \right\rceil \\ &= \lceil 28.838459 \rceil \\ &= 29 \text{ bits} \end{aligned} \tag{16}$$

The spur suppression requirement is the dominant consideration when selecting the depth of the sine/cosine lookup table. Of course, the table samples must be specified with an appropriate precision to support this value. As noted in an earlier section of this document, each sine/cosine table address bit contributes approximately 6 dB of spur suppression. To meet the -96 dB spur level, the number of address bits required is

$$\begin{aligned} B_{\Theta(n)} &= \left\lceil \frac{96}{6} \right\rceil \\ &= \lceil 16 \rceil \\ &= 16 \text{ bits} \end{aligned} \tag{17}$$

ıe table depth *N* is

$$N = 2^{B_{out}} = 2^{16} = 65536 \text{ samples} \quad (18)$$

ıe phase offset value *POFF* is determined as

$$\begin{aligned} POFF &= 2^{B_\phi} \cdot \frac{\phi}{2\pi} \\ &= \left\lceil 2^{29} \cdot \frac{\pi/4}{2\pi} \right\rceil \\ &= 67108864_{10} \end{aligned} \quad (19)$$

generate a 10.2 MHz output signal the phase increment ıuld be

$$\begin{aligned} \Delta\theta &= \frac{f_{out} 2^{B_{\theta(n)}}}{f_{clk}} \\ &= \frac{10.2\times10^6 \times 2^{29}}{120\times10^6} \\ &= \lceil 45634027.52 \rceil \\ &= 45634028 \end{aligned} \quad (20)$$

he *PINC* is configured to be programmable then the out- t frequency can be adjusted while the DDS is running. In ler for the DDS to be tuned to 8.2 MHz the new *PINC* can calculated to be

$$\begin{aligned} \Delta\theta &= \frac{f_{out} 2^{B_{\theta(n)}}}{f_{clk}} \\ &= \frac{8.2\times10^6 \times 2^{29}}{120\times10^6} \\ &= \lceil 36686178.98 \rceil \\ &= 36686178 \end{aligned} \quad (21)$$

write this new value to the *PINC* memory it must be pre- ıted on the *DATA* port along with an active *WE* and ɔropriate address on the *A* port. The *DATA* port will ɜume the same width as the *PACC* which was calculated ɔe 29 bits. The width of the DATA port is also reported on ge 4 (Figure 35) of the GUI. The PINC value represented a 29-bit binary value is

$9 = 00010001011111100100101100$

ıure 39 is a spectral plot of the 10.2 MHz output signal. serve that the highest spur meets the 96 dB spur sup- ıssion requirement.

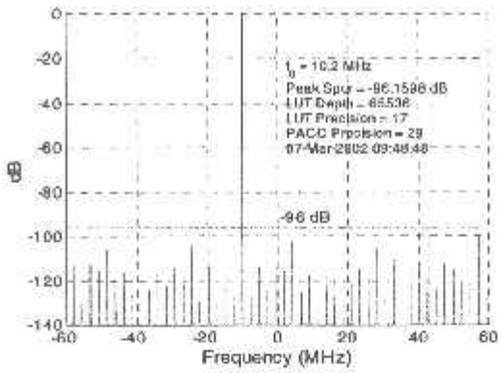


Figure 39: **Phase truncation DDS example 1. 10.2 MHz output tone. The highest spur is below the required 96 dB suppression value. The table depth is 65536 samples.**

In this example, the phase truncation DDS requires 16 of the 18Kb block RAMs for the lookup table storage. If a phase dithered DDS is used the table depth can be reduced to 16384 samples which reduces the memory requirement to 4 block RAMs. The output power spectrum for the preferred dithered DDS design is shown in Figure 40. Note that the minimum spur suppression has been achieved and the line structure of the spectrum shown in Figure 39 is no longer present.

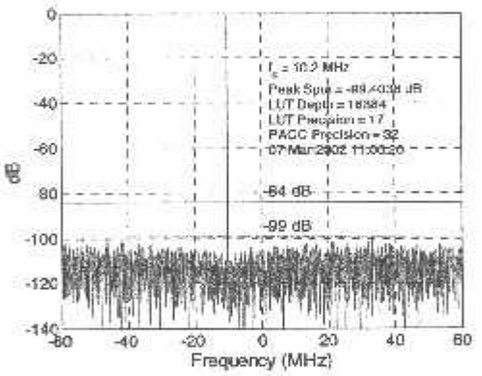


Figure 40: **Dithered DDS design for example 1. 10.2 MHz output tone.**

Using the dithered DDS, the highest spur, at -99.4038 dB, is well below the required 96 dB suppression value. If Virtex-II, Virtex-II Pro, Spartan-IIE, or Spartan-3 FPGAs are being used and two embedded multipliers are available the Taylor series corrected DDS can be used. While this architecture produces spurs at about -115 dB, which is much cleaner than the required -96 dB, the table depth is reduced to 4096 samples which fits in a single 18Kb block RAM. The output

ower spectrum for the Taylor series corrected DDS is hown in Figure 41.

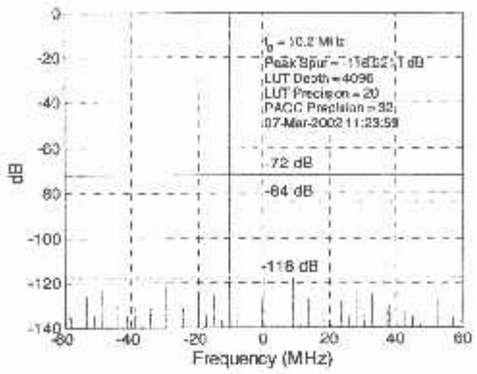


Figure 41: **Taylor series corrected DDS design for example 1, 1. 10.2Mhz output tone.**

sing the same requirements as before this example can e extended to use a multi-channel DDS. In doing so some f the calculations change slightly. Assuming a four channel DS the frequency resolution is defined by

$$\Delta f = \frac{f_{clk}}{\frac{channels}{2^{B_{\theta(n)}}}} \quad (22)$$

herefore, the bit-precision for the PACC register is

$$B_{\theta(n)} = \log_2\left\lceil \frac{\frac{f_{clk}}{channels}}{\Delta f} \right\rceil \quad (23)$$

$$= \left\lceil \log_2 \frac{\frac{120\times10^6}{4}}{0.25} \right\rceil$$

$$= \lceil 26.83845 \rceil$$

$$= 27 \text{ bits}$$

ssuming that the four tones to be generated are 8.2 MHz, 0.2 MHz, 12.2 MHz, and 14.2 MHz, the phase increment ould be

$$\Delta\theta = \frac{f_{out} 2^{B_{\theta(n)}}}{\frac{f_{clk}}{4}} \quad (24)$$

$$= \frac{f_{out} \times 2^{13}}{\frac{120\times10^6}{4}}$$

$$\Delta\theta_0 = 1120$$

$$\Delta\theta_1 = 1393$$

$$\Delta\theta_2 = 1666$$

$$\Delta\theta_3 = 1939$$

These four PINC values represented as 27-bit binary values are

$$\Delta\theta_0 = 000000000000000010001100000_2$$
$$\Delta\theta_1 = 000000000000000010101110001_2 \quad (25)$$
$$\Delta\theta_2 = 000000000000000011010000010_2$$
$$\Delta\theta_3 = 000000000000000011110010011_2$$

Figure 42 shows the power spectrum for the four-channel DDS implemented as a Taylor series corrected DDS.

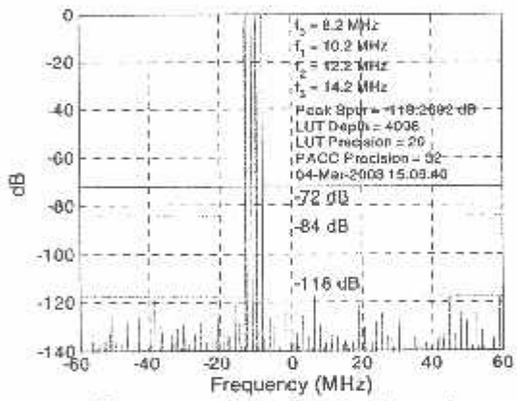


Figure 42: **Power spectrum for the four-channel DDS implemented as a Taylor series corrected DDS.**

DDS Design Example 2

Consider a DDS that is to satisfy the following requirements

frequency resolution $\Delta f = 0.3$ Hz
minimum spur suppression $S = 82$ dB
$f_{clk} = 80$ MHz
Quadrature output: $s(n) = e^{-j(\theta(n)+\varphi)} = \cos(\Theta(n)+\varphi) - j\sin(\Theta(n)+\varphi))$
where the phase offset $\varphi = -50$ degrees

The DDS GUI has been redesigned allowing the user to enter system-level parameters instead of low-level parameters such as the width of the phase accumulator. Because of this all of the requirements above can be entered into the GUI directly without having to calculate low-level core details. As an exercise this example will show some calculations that the core does automatically.

The field width of the phase accumulator (PACC in Figure 29), defines the DDS frequency resolution according to

$$\Delta f = \frac{f_{clk}}{2^{B_{\theta(n)}}} \tag{26}$$

Therefore, the bit-precision for the PACC register is

$$\begin{aligned} B_{\theta(n)} &= \log_2\left\lceil \frac{f_{clk}}{f} \right\rceil \\ &= \left\lceil \log_2 \frac{80 \cdot 10^6}{0.3} \right\rceil \\ &= \lceil 27.99046 \rceil \\ &= 28 \text{ bits} \end{aligned} \tag{27}$$

The spur suppression requirement is the dominant consideration when selecting the depth of the sine/cosine look-up table. Of course, the table samples must be specified with an appropriate precision to support this value. As noted in an earlier section of this document, each sine/cosine table address bit contributes approximately 6 dB of spur suppression. To meet the -82 dB spur level, the number of address bits required is

$$\begin{aligned} B_{\theta(n)} &= \left\lceil \frac{82}{6} \right\rceil \\ &= \lceil 13.666 \rceil \\ &= 14 \text{ bits} \end{aligned} \tag{28}$$

The table depth N is

$$N = 2^{B_{\theta(n)}} = 2^{14} = 16384 \text{ samples} \tag{29}$$

In this example, the required phase offset has been specified as a negative value. The phase offset value -50 degrees is the same as +310 degrees.

The phase offset value *POFF* is determined as

$$\begin{aligned} \text{POFF} &= N \cdot \frac{\varphi}{360} \\ &= \left\lceil 16384 \cdot \frac{310}{360} \right\rceil \\ &= 14336_{10} \end{aligned} \tag{30}$$

To generate an 8.4 MHz output signal, the phase increment would be

$$\begin{aligned} \Delta\theta &= \frac{f_{out} 2^{B_{\theta(n)}}}{f_{clk}} \\ &= \frac{8.4 \times 10^6 \times 2^{14}}{80 \times 10^6} \\ &= \lceil 1720.32 \rceil \\ &= 1720 \end{aligned} \tag{31}$$

If the DDS is configured such that the output frequency is programmable then the value of 1720 must be presented on the *DATA* port for writing to the *PINC* memory. The *DATA* port will assume the same width as the *PACC* which was calculated to be 28 bits. The *PINC* value represented as a 28-bit binary value is $\Delta\theta = 0001\,1010\,1110\,0001\,0100\,0111\,1011_2$.

Figure 43 is a spectral plot of the 8.4 MHz output signal. Observe that the highest spur meets the 82 dB spur suppression requirement.

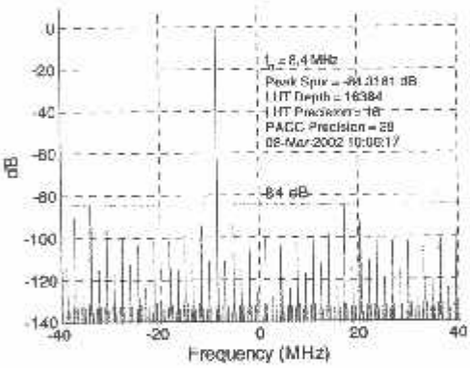


Figure 43: **Phase truncation DDS example 2. 8.4 MHz output tone. The highest spur is below the required 82 dB suppression value.**

The output power spectrum for the preferred dithered DDS design is shown Figure 44. In this case a 4096-deep table has been employed. Note that the minimum spur suppression has been achieved and the line structure of the spectrum shown in Figure 43 is no longer present. Since the desired spur suppression has been achieved while only consuming a single 18Kb block RAM the phase dithered DDS is the best area efficient solution, since moving to the Taylor series corrected architecture would require more resources.

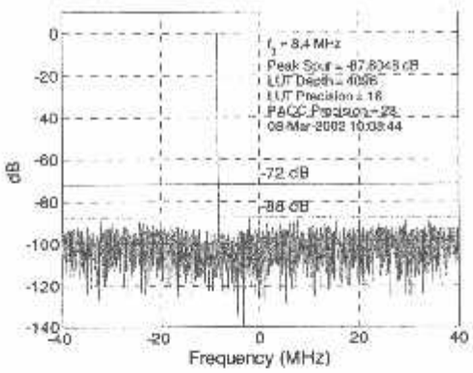


Figure 44: **Dithered DDS Design for Example 2. 8.4 MHz output tone. Using the dithered DDS the highest spur, at 88 dB, is below the required 82 dB suppression value. The design uses a table that is one quarter the depth of the phase truncation implementation.**

Ordering Information

This core may be downloaded from the Xilinx IP Center for use with the Xilinx CORE Generator system v7.1i and later. The CORE Generator system is bundled with the ISE Foundation software at no additional charge.

To order Xilinx software, please visit the Xilinx Silicon Xpresso Cafe or contact your local Xilinx sales representative.

Information on additional Xilinx LogiCORE modules is available on the Xilinx IP Center.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
03/15/03	1.0	Initial revision history.
05/21/04	2.0	Updated document to support Xilinx software v6.2I and Virtex-4 FPGA.
04/28/05	2.1	Updated support for Spartan-3E and Xilinx ISE software v7.1i.