

SIMULATOR *DIRECT DIGITAL SYNTHESIZER (DDS) 8 BIT* : CARA KERJA

Elan Djaelani, Riyo Wardoyo, Rico Dachlan

Pusat Penelitian Informatika LIPI

Kampus LIPI Gd. 20 Lt. 3

Jl. Sangkuriang Bandung 40135 – INDONESIA

Email: *elan@informatika.lipi.go.id*

Diterima : 07 Februari 2014

Disetujui : 28 Maret 2014

Dipublikasikan : Maret 2014

ABSTRAK

Penggunaan *Direct Digital Synthesizer (DDS)* telah banyak pada berbagai bidang, LIPI telah menggunakannya pada radar dan jammer, sehingga minat belajar *DDS* meningkat pesat. Namun, harga *DDS* masih relatif mahal, khususnya untuk siswa. Untuk alasan ini, penting untuk mengembangkan simulator untuk *DDS* sebagai alternatif yang lebih murah untuk belajar tentang *DDS*. Tulisan ini membahas cara kerja simulator *DDS* 8bit. Simulator *DDS* 8 bit dibuat memakai software LabView oleh National Instrumentation (NI). Simulator ini terdiri 2 bagian yaitu : *frontpanel* dan *diagramblock*. *Frontpanel* mempunyai beberapa tabel, yaitu : *accumulator counter 8 bit*, *accumulator counter boolean array lsb msb* , *lookup memori adres lsb msb*, dan *lookup memori adres* .Hasil pembacaan tabel tabel tersebut ditempatkan pada bagian pengukuran. *Diagramblock* mempunyai 2 buah sub blok :*lookup tabel sinus* dan rangkaian *accumulator*. *Lookuptabel sinus* mempunyai 16 sample sinus dalam satu periode , dimana beda sudut antara dua sampel adalah 0,393 radian. Rangkaian *Accumulator* terdiri dari : *multiply*, *pengubah number to boolean array*, *array subset*, *Pengubah boolean array to number* dan *index array*. Perhitungan output rangkaian rangkaian tersebut hasilnya terdapat pada bagian hasil dan pembahasan. Dengan membandingkan pembacaan tabel dan perhitungan output rangkaian rangkaian pada *diagram block* maka dapat menyimpulkan rangkaian tersebut menghasilkan output yang benar. Hal tersebut diatas sekaligus menerangkan cara kerja dari simulator.

Kata kunci: cara kerja , simulator *DDS* 8bit

ABSTRACT

The use of direct digital synthesizer (DDS) has been much in various fields. LIPI has used it on radar and the jammer, so that learning DDS interest soared. But DDS prices still relatively expensive Especially for students. For this reason, it is important to develop a simulator for DDS as a cheaper alternative to learning about DDS. This writing discuss ways of working simulator DDS 8bit. 8-bit DDS Simulator created using LabView software by National Instrumentation (NI). The Simulator consists of 2 parts: the frontpanel and diagramblock. *Frontpanel* have several table, namely : *accumulator counter 8 bit*, *accumulator counter boolean array lsb msb* , *lookup memori adres lsb msb*, dan *lookup memori adres* . The results of the reading of those tables placed on the measurement. *Diagramblock* have 2 item sub block :*lookup tabel sinus* and sequence *accumulator*. *Lookuptabel sinus* have 16 sample of sinus in a periode , where the angular difference between the two samples was 0,393 radians. Sequence of *Accumulator* consist of : *multiply*, modifier *number to boolean array*, *array subset*, Modifier *boolean array to number* and *index array*. The output circuit calculation the series as a result there is the on the part of the results and discussions. By comparing the reading table and by computation output of the series on a diagram block then can conclude the series of producing such an output that right. It mentioned above at once explained that the working mechanism of the simulator.

Keywords: ways of working, DDS 8bit simulator

PENDAHULUAN

Karena banyaknya penggunaan *DDS* pada berbagai bidang ,maka minat belajar *DDS* meningkat pesat. Namun, harga *DDS* masih relatif mahal, khususnya untuk siswa. Untuk alasan ini, penting untuk mengembangkan simulator untuk *DDS* sebagai alternatif yang lebih murah untuk belajar tentang *DDS*. *Simulator DDS 8 bit* yang telah dibuat NI, berupa program LabView [1]. Setelah program dijalankan kita dapat melihat *diagramblock* dan *frontpanel*,seperti pada gambar 1 dan gambar 2. Pada *front panel* dapat mengisi besar *frekuensi*, pada percobaan ini diset 3 Hz.Lalu *click* tanda panah *run*,maka akan keluar hasil simulasi.Hasil simulasi berupa :display berupa LED,tabel,display bentuk gelombang. Pada *front panel* terdapat tabel *accumulator counter 8 bit*, *accumulator counter boolean array lsb msb*, *lookup memori adres lsb msb*, *lookup memori adres* .Pembacaan tabel diatas hasilnya ditempatkan pada bagian pengukuran. *Diagramblock* simulator pada gambar 4.Ia terdiri dari komponen: *multiply*, pengubah *number to boolean array*, *array subset*, dan pengubah *boolean array to number*. Perhitungan output rangkaian rangkaian tersebut terdapat pada bagian hasil dan pembahasan. Outline tulisan ini, pertama pendahuluan, digital signal prosesi, pengenalan software simulasi, pengenalan *DDS*, simulasi *DDS*, pengukuran, hasil dan analisis, kesimpulan dan daftar pustaka.

Digital Signal Prosesi

Pada buku digital sinyal prosesi persamaan sinyal sinus seperti persamaan (1) [2]. Kontinyu time domain dari sinyal sinus:

$$x(t) = \sin(2\pi f_0 t) \dots\dots\dots(1)$$

Dimana:

f_0 =frekuensi dasar.

Sinyal sinus seperti persamaan (1) disampling setiap $t = t_s$, maka terjadi persamaan diskrit dari sinus tersebut.

$$x(0) = \sin(2\pi f_0 0t_s), \text{ nilai sampling ke-0.}$$

$$x(1) = \sin(2\pi f_0 1t_s), \text{ nilai sampling ke-1.}$$

$$x(2) = \sin(2\pi f_0 2t_s), \text{ nilai sampling ke-2.}$$

$$x(n) = \sin(2\pi f_0 nt_s), \text{ nilai sampling ke-n.}$$

Dari sifat sinyal sinus , maka

$$\sin(\phi) = \sin(\phi + 2\pi n) \dots\dots\dots(2)$$

Dari persamaan (2) maka persamaan diskrit dapat diubah kebentuk pers (3), lalu diubah menjadi pers (4).

$$x(n) = \sin(2\pi f_0 nt_s) = \sin(2\pi f_0 nt_s + 2\pi n) = \sin(2\pi(f_0 + \frac{m}{nt_s})nt_s) \dots\dots\dots(3)$$

$$x(n) = \sin(2\pi(f_0 + \frac{k}{t_s})nt_s)$$

$$x(n) = \sin(2\pi f_0 nt_s) = \sin(2\pi(f_0 + kf_s)nt_s) \dots\dots\dots(4)$$

Pada *Digital Sinyal Prosesing* kita dapat merekam sample-sample sinyal sinus pada memori. Prosesnya melalui tahapan pertama melakukan sampling gelombang sinus, kedua melakukan kuantisasi, ketiga mengkodekan pada digital dan terakhir menyimpan pada memori. Pada *DDS* kebalikannya, pada chip ini dapat membaca sampel-sampel gelombang sinus dari *lookup tabel sinus*, lalu mengirim ke DAC, melewati pada *lowpass filter*, maka akan terbentuk sinyal gelombang sinus *DDS* diprogram oleh prosesor atau komputer, menghasilkan gelombang sinus dengan stabilitas dan akurasi *frekuensi* yang sangat tinggi.

PENGENALAN SOFTWARE SIMULASI

LabVIEW (singkatan dari *Laboratory Virtual Instrumentation Engineering Workbench*) adalah perangkat lunak komputer untuk pemrosesan dan visualisasi data dalam bidang akuisisi data, kendali instrumentasi serta automasi industri yang pertama kali dikembangkan oleh perusahaan NI pada tahun 1986. Perangkat lunak ini dapat dijalankan pada sistem operasi Linux, Unix, Mac OS X dan Windows [3]. Ada beberapa simbol LabView yang dipergunakan pada makalah ini, diantaranya :

1. *Divide*
2. *Multiply*
3. *Sine*
4. *To Unsigned Byte Integer*
5. *Number To Boolean Array*
6. *Array Subset*
7. *Boolean Array To Number*
8. *Index Array*

Semua simbol tersebut diatas ada pada gambar 3 dan gambar 4 pada makalah ini. *Divide* adalah menghitung membagi dari input input. Input input dapat berupa bilangan skalar, array atau kumpulan dari bilangan, array dari kumpulan dari bilangan. *Multiply* adalah hasil perkalian dari input input. Input input dapat berupa bilangan skalar, array atau kumpulan dari bilangan, array dari kumpulan dari bilangan. *Sine* adalah icon menghitung sine dari x, x dalam radians. *To unsigned long integer* yaitu icon yang mengubah number menjadi 32 bit integer.

Number to Boolean array yaitu icon yang dapat mengubah fixed number atau integer menjadi array. *Array subset* yaitu icon yang berfungsi untuk mengambil sebagian array (sub array) dari array total. Angka 4 berarti sub array dimulai dari index 4 sepanjang 4 element (arti angka 4) disinilah proses penentuan alamat pada look up table terjadi.

Boolean array to number.vi yaitu icon yang berfungsi mengubah Boolean array menjadi 32 bit integer. *Index array* yaitu icon yang berfungsi mengambil nilai pada sub array sesuai index yang diberikan. *index array* berfungsi mengambil nilai pada look up table sesuai alamat yang diberikan.

Pengenalan DDS

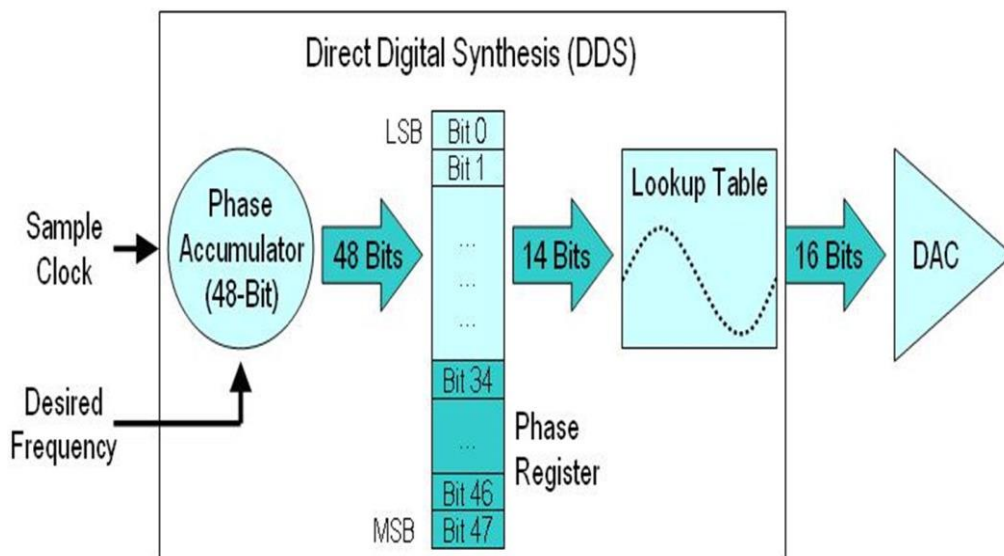
DDS secara luas digunakan untuk aplikasi telekomunikasi seperti RADAR and JAMMER[4][5]. Lembaga Ilmu Pengetahuan Indonesia (LIPI) telah mengembangkan sebuah radar kelautan dan radar pengawasan, yang digunakan untuk mengukur jarak dan

kecepatan kapal dan membimbing mereka.

ELAN DJAELANI DKK : SIMULATOR DIRECT DIGITAL SYNTHESIZER (DDS) 8 BIT : CARA KERJA

Radar ini adalah sebuah radar *Frekuensi Modulation Continuous Wave* (FMCW). Ini adalah sebuah generasi baru radar yang terus-menerus mengirimkan sinyal. Hal ini berbeda dari radar pulsa *konvensional* yang mengirimkan pulsa secara berkala. Untuk radar FMCW, pulsa digantikan oleh sinyal yang dimodulasi dengan sinyal, yang disebut pembawa, yang memiliki periode sama dengan periode pulsa. Pembawa ini dapat berupa sinusoid, gergaji, segitiga, dll Dalam radar, DDS ini digunakan untuk menghasilkan sebuah sinyal yang berdecit. Sinyal yang berdecit adalah sinyal di mana frekuensi perubahan (kenaikan atau penurunan) sesuai dengan waktu. Sinyal ini dapat dihasilkan oleh modulasi sinyal dengan sinyal gergaji. LIPI juga telah mengembangkan sebuah jammer yang beroperasi di frekuensi militer. Jammer adalah pemancar radio yang digunakan untuk mengganggu komunikasi radio. Aplikasi ini juga menerapkan DDS.

DDS [6] adalah teknik untuk menghasilkan sinyal keluaran acak dari referensi clock tunggal yang tetap. DDS biasanya digunakan untuk menghasilkan sinyal, sebagai osilator, modulator, fungsi generator, dll dalam sistem telekomunikasi. Di DDS, clock referensi dibagi dengan faktor-faktor skala tetap dalam jumlah yang dapat diprogram tuning biner. Jumlah tuning adalah antara 24 hingga 48 bit. Karena alasan ini, DDS dapat menghasilkan sebuah resolusi tinggi dari frekuensi tuning. DDS juga telah memiliki kecepatan tinggi dan kinerja tinggi dari sebuah converter digital ke analog(D/A).



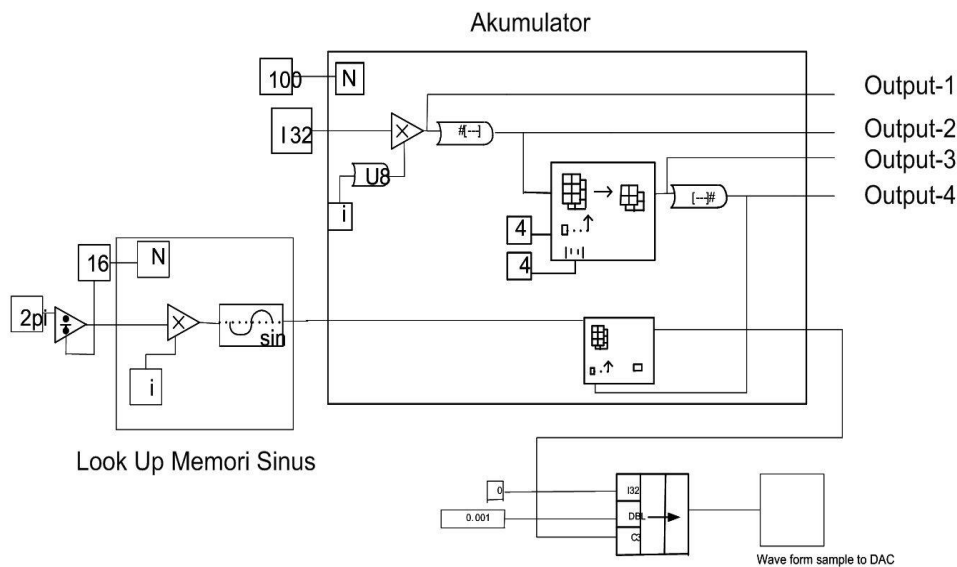
Gambar 1. Arsitektur DDS.[7]

Arsitektur DDS dapat dilihat pada gambar 1. Terdiri 3 bagian utama : Phase Accumulator, Phase Register dan Lookup Table.

Simulator DDS

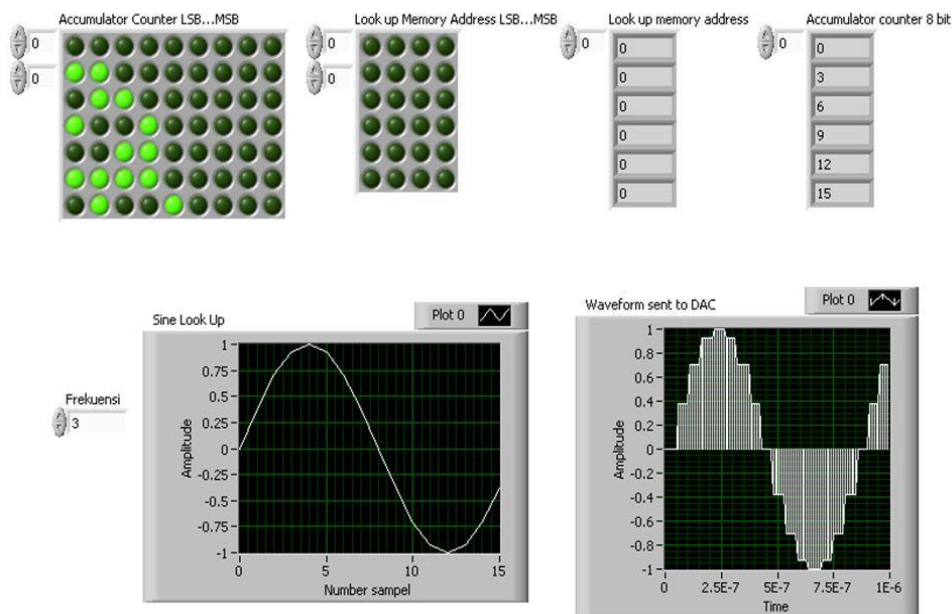
Blok diagram Simulator DDS buatan NI

Dalam bagian ini, simulasi DDS dari NI adalah secara singkat dijelaskan sbb.[1].



Gambar 2. Diagram block simulasi DDS dari NI

Gambar 2 menunjukkan *diagramblock* simulator DDS. Simulasi dari NI menggunakan 8 bit data. DDS pada dasarnya terdiri dari lookup memori, accumulator, dan counter. Gambar 3 adalah Front Panel DDS 8 bit. Simulator diseting frekuensinya sama dengan 3 Hz.

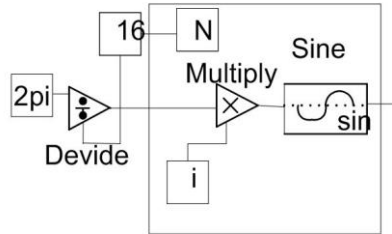


Gambar 3. Front panel DDS 8 bits

Frontpanel simulasi dapat dilihat pada Gambar 3. yang terdiri dari accumulator counter dan alamat memori lookup (dalam biner dan desimal).

Cara kerja *Lookup Memori*

Blok diagram *Lookup Memori*

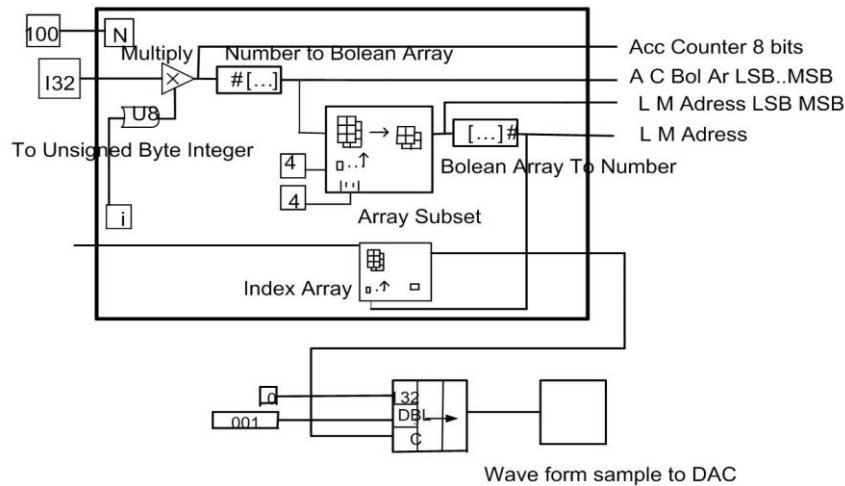


Gambar 4. Diagram block *Lookup Memori*

Gambar 4. adalah suatu *Forloop* dari *Lookup Memori* ,dengan $N=16$ dengan $i=0,1,\dots,15$. Sudut per sampel = $2\pi/16=0,393$ radian. Pada tabel 0 terlihat; kolom 1 nomor, kolom 2 sudut persampel, kolom 3 In Sine (dalam radian), dan kolom 4 Out Sine. Komponen *Lookup Tabel Sinus* terdiri:*Divide, Multiply, Sine*. Keluaran *Divide* berupa sudut persampel yaitu : $2\pi/16=0,393$ radian. Keluaran *Multiply*= $i \times 0,393$. Dimana $i=0,1,\dots,15$. Keluaran *Sine*= $\sin (i \times 0,393)$. Nilai nilai diatas seperti pada Tabel 0,bab Pengukuran.

Cara kerja *Accumulator*

Gambar 5 adalah sebuah *ForLoop* dari *Accumulator*. Komponen komponennya terdiri dari:*To Unsigned Byte Integer, Multiply, Number To Boolean Array, Array Subset, Boolean Array to Number dan Index Array*.



Gambar 5. *Accumulator*

Rangkaian *Multiply* mengalikan Frekuensi yang diinginkan dengan bilangan i , dimana $i=0,\dots,99$. Output *Multiply* seperti pada table 1. Hasil perkalian ini yang berupa bilangan, masuk ke komponen yang mengubah bilangan ke array Boolean (*Number to Boolean Array*). Output komponen ini seperti pada table 2. Output komponen (*Number to Boolean Array*) masuk ke rangkaian *Array Subset*.

Rangkaian ini berfungsi untuk mengambil sebagian array (*sub array*) dari array total. Angka 4 berarti *sub array* dimulai dari index 4 sepanjang 4 element (arti angka 4) disini adalah proses penentuan alamat pada look up table terjadi. Output *Array Subset* seperti tabel 3. Output *Array Subset* masuk kerangkaian pengubah *Boolean Array to Number*. Output seperti pada tabel 4. Output rangkaian pengubah *Boolean Array to Number* masuk ke rangkaian *Index Array*. Rangkaian ini berfungsi mengambil nilai pada sub array sesuai index yang diberikan. Rangkaian ini berfungsi mengambil nilai pada *look up table* sesuai alamat yang diberikan.

HASIL PENGUKURAN

Pembacaan Tabel *Look Up Memori*

Yang dimaksudkan pengukuran disini adalah pembacaan display *Front Panel*. Output *Look Up Tabel Memori* seperti pada tabel-0. Kolom 1 adalah nomor, kolom 2 adalah sudut persampel, kolom 3 adalah In Sine dalam radian, kolom 4 adalah Out Sine.

Tabel 0. *Lookup Tabel sinus*

No	Sudut/sampel	In Sine	Out Sine
0	0,393	0	0
1	0,393	0,393	0,3824995
2	0,393	0,785	0,7068252
3	0,393	1,18	0,9236508
4	0,393	1,57	0,9999997
5	0,393	1,96	0,92426
6	0,393	2,36	0,7079509
7	0,393	2,75	0,3839706
8	0,393	3,14	0,0015927
9	0,393	3,53	-0,3810275
10	0,393	3,93	-0,7056977
11	0,393	4,32	-0,9230393
12	0,393	4,71	-0,9999971
13	0,393	5,10	-0,9248668
14	0,393	5,50	-0,7090748
15	0,393	5,89	-0,3854406

Pembacaan display *Front Panel: Accumulator Counter 8 bit*

Pembacaan display *Accumulator Counter 8 bit* seperti pada table 1.

Tabel 1. *Accumulator 8 bit (telah disederhanakan)*

i	Out	i2	Out3
1	3	54	162
6	18	59	177
11	33	64	192
16	48	70	210
22	66	75	225
27	81	80	240
32	96	86	258
38	114	91	273
43	129	96	288
48	144		

Pembacaan display Front Panel: Accumulator Counter Boolean Array LSB MSB

Pembacaan display Accumulator Counter Boolean Array LSB MSB seperti pada table 2.

Tabel 2. *Number to Boolean Array (telah disederhanakan)*

i	Out	i2	Out3
1	"110000000"	48	"000010010"
6	"010010000"	54	"010001010"
11	"100001000"	59	"100011010"
16	"000011000"	64	"000000110"
22	"010000100"	70	"010010110"
27	"100010100"	80	"000011110"
32	"000001100"	86	"010000001"
38	"010011100"	91	"100010001"
43	"100000010"	96	"000001001"

Pembacaan display Front Panel: Pembacaan LookUp Memori Adres Boolean Array LSB MSB

Pembacaan display Look Up Memori Adres Boolean Array LSB MSB. seperti pada tabel 3.

Tabel 3. *Lookup Memori Adres Boolean Array (telah disederhanakan)*

Input	output	input	output
"110000000"	"0000"	"010001010"	"0101"
"010010000"	"1000"	"100011010"	"1101"
"100001000"	"0100"	"000000110"	"0011"
"000011000"	"1100"	"010010110"	"1011"
"010000100"	"0010"	"100001110"	"0111"
"000001100"	"0110"	"000011110"	"1111"
"010011100"	"1110"	"010000001"	"0000"
"100000010"	"0001"	"100010001"	"1000"
"000010010"	"1001"	"000001001"	"0100"

Pembacaan display Front Panel : Pembacaan LookUp Memori Adres

Pembacaan display Look Up Memori Adres seperti pada table 4.

Tabel 4. *Look Up Memori Adress (telah disederhanakan)*

i	output	i	output
"0000"	"0"	"1001"	"9"
"1000"	"1"	"0101"	"10"
"0100"	"2"	"1101"	"11"
"1100"	"3"	"0011"	"12"
"0010"	"4"	"1011"	"13"
"1010"	"5"	"0111"	"14"
"0110"	"6"	"1111"	"15"
"1110"	"7"	"0000"	"0"
"0001"	"8"	"1000"	"1"
"1001"	"9"	"0100"	"2"

HASIL DAN ANALISIS

Mengitung input (dalam radian), sin (input dalam radian) dengan menggunakan kalkulator.

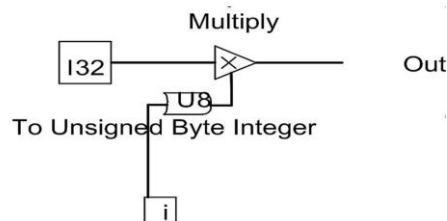
Tabel 5. *Out Sine*

In	Out
0	0
0,393	0,3824
0,785	0,7068
1,18	0,9246
1,57	0,9999
1,96	0,9252
2,36	0,7044
2,75	0,3816
3,14	0,0015
3,53	-0,3787
3,93	-0,7092
4,32	-0,9239
4,71	-0,9999
5,10	-0,9258
5,50	-0,7055
5,89	-0,3831

Hasil perhitungan table 5 sin out dibandingkan dengan pengukuran table 0, mempunyai hasil sama.

1. Menghitung perkalian

Memilih frekuensi yang diinginkan adalah 3 Hz. Rangkain multiply seperti gambar 6.



Gambar 6. Multiply

Hasil perhitungan ada pada table 6.

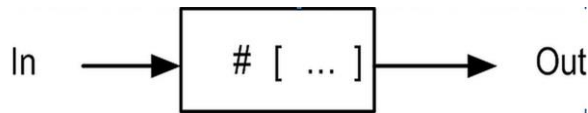
Tabel 6. *Out Multiply (telah disederhanakan)*

i32	i	Out	i322	i3	Out4
3	1	3	3	54	162
3	6	18	3	59	177
3	11	33	3	64	192
3	16	48	3	70	210
3	22	66	3	75	225
3	27	81	3	80	240
3	32	96	3	86	258
3	38	114	3	91	273
3	43	129	3	96	288
3	48	144			

Hasil perhitungan pada table 6. output *multiply* (perkalian) dibandingkan dengan Tabel 1. *Accumulator 8 bit*, mempunyai hasil sama.

2. Mengubah *Number To Boolean Array*

Mengubah *Number to Boolean Array* seperti pada gambar 7.



Gambar 7. Number To Boolean Array

Hasil mengubah *Number to Boolean Array* seperti pada table 7.

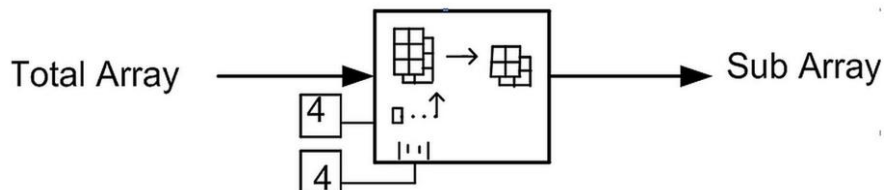
Tabel 7. *Number To Boolean Array* (telah disederhanakan)

In	Out	In	Out
3	"110000000"	162	"010001010"
18	"010010000"	177	"100011010"
33	"100001000"	192	"000000110"
48	"000011000"	210	"010010110"
66	"010000100"	225	"100001110"
81	"100010100"	240	"000011110"
96	"000001100"	258	"010000001"
114	"000110000"	273	"100010001"
129	"000110000"	288	"000001001"
144	"111001000"		

Hasil perhitungan pada table 7. *Number To Boolean Array* dibandingkan dengan Tabel 2. *Number to Boolean Array*, mempunyai hasil sama.

3. Output Array Subset

Rangkaian *Array Subset* seperti pada gambar 8.



Gambar 8. Array Subset

Input dan output *Array Subset* seperti pada table 8 (telah disederhanakan)

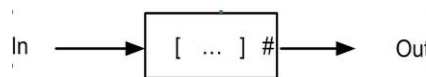
Table 8. *Input dan output Array Subset*

input	output	input	output
"11000000"	"0000"	"010001010"	"0101"
"010010000"	"1000"	"100011010"	"1101"
"100001000"	"0100"	"000000110"	"0011"
"000011000"	"1100"	"010010110"	"1011"
"010000100"	"0010"	"100001110"	"0111"
"100010100"	"1010"	"000011110"	"1111"
"000001100"	"0110"	"010000001"	"0000"
"010011100"	"1110"	"100010001"	"1000"
"100000010"	"0001"	"000001001"	"0100"
"000010010"	"1001"		

Hasil perhitungan pada table 8. Input dan output *Array Subset* dibandingkan dengan Tabel 3. Lookup Memori Adres Bolean Array, mempunyai hasil sama.

4. Bolean Array To Number

Rangkaian pengubah Bolean Array To Number seperti pada gambar 9.



Gambar 9. Boolean array to Number

Input dan output pengubah Bolean Array To Number seperti pada table 9.

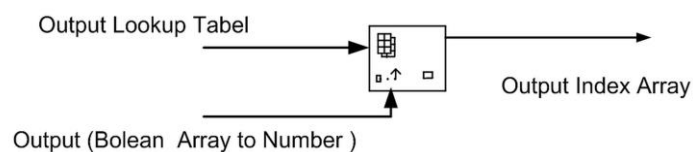
Tabel 9. *Boolean Array to Number (telah disederhanakan)*

input	output	input	output
"0000"	0	"1001"	9
"1000"	1	"0101"	10
"0100"	2	"1101"	11
"1100"	3	"0011"	12
"0010"	4	"1011"	13
"1010"	5	"0111"	14
"0110"	6	"1111"	15
"1110"	7	"0000"	0
"0001"	8	"1000"	1

Hasil perhitungan pada Tabel 9. Bolean Array to Number dibandingkan dengan table input dan output Bolean Array to Number, mempunyai hasil sama.

5. Index Array

Rangkaian index Array speti pada gambar 10.



Gambar 10. index Array

Input dan output Index Array seperti pada table 10.

Tabel 10. *Input dan output Index Array*

Input	Output	Input2	Output3
0	0	8	0.0015927
1	0,3824995	9	-0.3810275
2	0,7068252	10	-0.7056977
3	0.9236508	11	-0.9230393
4	0.9999997	12	-0.9999971
5	0.92426	13	-0.9248668
6	0.7079509	14	-0.7090748
7	0.3839706	15	-0.3854406

Output Index Array sama dengan pembacaan table sine. Hasil perhitungan pada Tabel 10. Input dan output Index Array dibandingkan dengan Tabel 0. Lookup Tabel sinus, mempunyai hasil sama.

KESIMPULAN

Tabel tabel pada bagian pengukuran adalah hasil pembacaan pada tabel tabel yang ada pada *Front panel*. Tabel tabel yang berada pada hasil dan pembahasan, adalah merupakan hasil perhitungan rangkaian pada *diagram block*.

Hasil tabel dari Pengukuran dibandingkan dengan tabel Hasil dan Pembahasan mempunyai nilai sama, maka rangkaian pada *diagram block* sesuai dengan output simulator.

Cara kerja rangkaian pada *diagram block* adalah merupakan analisis dari simulasi DDS 8 bit.

DAFTAR PUSTAKA

- [1] National Instrumentasi: "Simulasi DDS 8 bit".
- [2] Richard G. Lyons, "Understanding Digital Signal Processing", Adison Wesley, 1997. Halaman 25.
- [3] Lisa K. Wells, Student Edition User's Guide, LabVIEW, National Instruments, 1994.
- [4] Elan Djaelani, "Perangkat Jamming Hasil Modifikasi Berbasis Direct Digital Synthesizer", Seminar Nasional Ilmu Pengetahuan Teknik Bandung, 28-29 November 2012.
- [5] Purwoko Adhi, "Pembangkitan chirp untuk radar FMCW berbasis DDS", Jurnal Elektronika No.2, Vol.11, Juli Desember 2011.
- [6] National Instruments, *Understanding Direct Digital Synthesis (DDS)*.
- [7] A Technical Tutorial on Digital Sinyal Synthesis, Analog Devices, 1999.