PERANCANGAN NON-RESTORING DIVIDER 8-BIT DENGAN TEKNOLOGI 180NM MENGGUNAKAN PERANGKAT LUNAK ELECTRIC

Satya Danniswara[*)], Munawar Agus Riyadi, dan Darjat

Departemen Teknik Elektro, Universitas Diponegoro
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

[*)]*E-mail: satyadanniswara@yahoo.co.id*

Abstrak

Operasi pembagian merupakan salah satu operasi penting yang ada pada blok Arithmetic Logic Unit (ALU). Meskipun operasi pembagian lebih jarang digunakan jika dibandingkan dengan penjumlahan dan perkalian, lamanya waktu yang dibutuhkan untuk menyelesaikan operasi ini menyebabkan banyak energi yang terbuang. Untuk mengatasi permasalahan tersebut, terdapat beberapa teknik yang dapat dilakukan, salah satunya adalah dengan memilih algoritma yang tepat sehingga operasi yang dilakukan juga semakin cepat. Pada penelitian ini, dirancang sebuah divider menggunakan teknologi 180nm dengan algoritma non-restoring. Dalam penerapannya, digunakan perangkat lunak Electric untuk merancang layout dan LTspice untuk menguji fungsional serta melakukan pengukuran timing delay. Dari perancangan yang dilakukan, didapati divider ini memiliki luas sebesar 0,04mm2, propagation delay time sebesar 3,938ns, dan area coverage sebesar 40,115%.

Kata kunci: divider, 180nm, algoritma non-restoring

Abstract

Division operation is one of the most important operations in Arithmetic Logic Unit (ALU) block. Although division operation is an infrequent operation compared to addition and multiplication, its longer latency makes division operation dissipate a significant amount of energy. There are some techniques to solve the problem, one of them is a precise algorithm choosing so the operation will be done faster. In this research, a divider using 180nm technology based on non-restoring algorithm is designed. In application, this research used Electric to design the layout and LTspice to do functional test and to measure timing delay. As a result, this divider has 0,04mm2 for area, 3,938ns for propagation delay time, and 40,115% for area coverage.

Keywords: divider, 180nm, non-restoring algorithm

1. Pendahuluan

Perancangan rangkaian terpadu (integrated circuit, disingkat IC) memiliki peran penting dalam menjawab tuntutan zaman yang menginginkan teknologi serba lebih baik. Peningkatan kecepatan akses, pengurangan konsumsi daya, dan pengecilan ukuran adalah beberapa upaya yang dapat dilakukan untuk memperoleh teknologi terbaik. Upaya-upaya tersebut berkaitan erat dengan perancangan rangkaian terpadu, lebih lanjut disebut teknologi IC khususnya Very-large-scale integration (VLSI) [4].

Salah satu penerapan VLSI adalah pada prosesor. Untuk dapat melakukan pemrosesan data, prosesor terdiri dari beberapa blok utama. Salah satu blok terpenting dari prosesor di antaranya adalah Arithmetic Logic Unit (ALU). ALU merupakan blok yang berfungsi untuk melakukan operasi perhitungan seperti penjumlahan, pengurangan, pengalian, pembagian, penggeseran bit, dan lain-lain [5].

Operasi pembagian merupakan salah satu operasi dasar yang ada pada blok ALU selain operasi penjumlahan, pengurangan, dan perkalian. Meskipun operasi pembagian lebih jarang digunakan jika dibandingkan dengan penjumlahan dan perkalian, lamanya waktu yang dibutuhkan untuk menyelesaikan operasi ini membuat banyak energi yang terbuang [1]. Untuk mengatasi permasalahan tersebut, terdapat beberapa teknik yang dapat dilakukan, salah satunya adalah dengan memilih algoritma yang tepat sehingga operasi yang dilakukan juga semakin cepat.

Terdapat banyak algoritma yang dapat digunakan untuk melakukan proses pembagian. Penelitian yang dilakukan oleh Siba Kumar Panda dan Arati Sahu dengan judul “A Novel Vedic Divider Architecture with Reduced Delay for

VLSI Applications" menggunakan metode Vedic mathematic-Parvatya Sutra. Penelitian ini disimulasikan menggunakan Xilinx ISE Simulator dan diimplementasikan pada virtex4 FPGA divais XC4VLX15 [2]. Penelitian lain dilakukan oleh C.-L.Wey dan C.-P.Wang dengan judul "Design of a Fast Radix-4 SRT Divider and its VLSI Implementation" yang menggunakan metode estimasi [3]. Selain itu, terdapat pula perancangan divider dengan menggunakan algoritma non-restoring oleh Pradeep Nair, Dhireesha Kudithipudi et al, namun tidak dijelaskan secara rinci seperti apa kinerja divider tersebut [1].

Penelitian ini bertujuan untuk merancang sebuah divider menggunakan teknologi 180nm dengan algoritma non-restoring melalui perangkat lunak Electric. Rancangan tersebut kemudian diuji secara fungsional dan diukur seberapa besar luas, delay, dan area coverage-nya.

2. Metode

2.1. Pembagian Bilangan Biner Algoritma *Non-restoring*

Terdapat beberapa metode atau algoritma yang dapat digunakan untuk melakukan operasi pembagian pada bilangan biner. Secara garis besar, algoritma tersebut dikelompokkan menjadi dua berdasarkan operasi perulangannya (iterative operator). Algoritma pertama menggunakan operasi perulangan pengurangan, sementara algoritma kedua menggunakan operasi perulangan perkalian [4]. Algoritma non-restoring termasuk ke dalam algoritma jenis pertama.

Seperti telah disebutkan sebelumnya, terdapat istilah operasi perulangan. Maksud operasi perulangan di sini adalah, operasi tersebut (dalam algoritma non-restoring berarti pengurangan) dilakukan berulang-ulang hingga mencapai hasil akhir. Untuk lebih jelasnya, berikut merupakan algoritma non-restoring yang digunakan untuk perancangan divider ini [5].

1. Mulai.
2. Tentukan nilai pembagi (divisor) dan yang dibagi (dividend).
3. Ubah divisor menjadi bilangan 2's complement-nya sehingga didapat nilai divisor negatif.
4. Susun nilai partial remainder awal, yaitu bilangan 0 sebanyak n-bit diikuti dengan bilangan dividend-nya (n merupakan besarnya bit pada divisor dan dividend)
5. Geser nilai partial remainder ke kiri sebesar 1 bit
6. Jumlahkan nilai partial remainder yang sudah digeser ke kiri sebesar 1 bit (langkah 5) dengan divisor yang sudah diubah menjadi bilangan 2's complement-nya (langkah 2).
7. Jika hasil penjumlahan tidak menghasilkan carry out, maka nilai divisor yang digunakan pada penjumlahan selanjutnya adalah nilai divisor positif dan quotient yang diberikan adalah '0'. Jika hasil penjumlahan menghasilkan carry out, maka nilai divisor yang digunakan pada penjumlahan selanjutnya adalah nilai divisor negatif dan quotient yang diberikan adalah '1'.
8. Ulangi langkah 5 sampai langkah 7 sebanyak n kali hingga didapatkan hasil akhir berupa n bit hasil bagi (quotient) dan n bit sisa bagi (remainder).
9. Selesai.

Perhatikan bahwa terdapat penjumlahan dengan bilangan 2's complement pada algoritma tersebut. Penjumlahan dengan bilangan 2's complement tak lain merupakan operasi pengurangan, salah satu ciri dari algoritma jenis pertama.

Setelah memahami bagaimana alur/algoritma dari operasi pembagain yang dilakukan, pengekstraksian ke dalam bentuk blok-blok apa saja yang dibutuhkan dapat dilakukan. Pada subbab selanjutnya akan dijelaskan mengenai perancangan blok-blok dan unit-unit dasar yang dibutuhkan. Perancangan blok-blok dan unit-unit dasar ini dirangkum dalam satu subbab yaitu perancangan standard cell.

2.2. Perancangan Standard Cell

Cell dapat diartikan sebagai unit atau blok yang dirancang. Perancangan standard cell sendiri dilakukan agar setiap cell yang dibuat mempunyai tinggi-lebar yang seragam, lebar VDD-GND yang sama, dan penggunaan metal dengan fungsi yang sama [6]. Perancangan standar cell dimulai dari tahap skematik hingga tahap layout.

Seperti yang telah dijelaskan sebelumnya, terdapat unit-unit dasar dan blok-blok yang dibutuhkan guna merancang divider. Unit-unit dasar dapat berfungsi sebagai penyusun blok maupun digunakan langsung untuk keperluan perancangan divider. Unit-unit dasar yang dibutuhkan terdiri atas beberapa gerbang logika dasar, buffer, tristate, XOR, multiplexer, D flip-flop tanpa reset, dan D flip-flop dengan reset.

Setelah unit-unit dasar yang dirancang sesuai secara fungsional, maka dilanjutkan dengan perancangan blok. Dari algoritma yang telah dijelaskan pada subbab 2.1, divider yang dirancangan memerlukan blok-blok sebagai berikut.

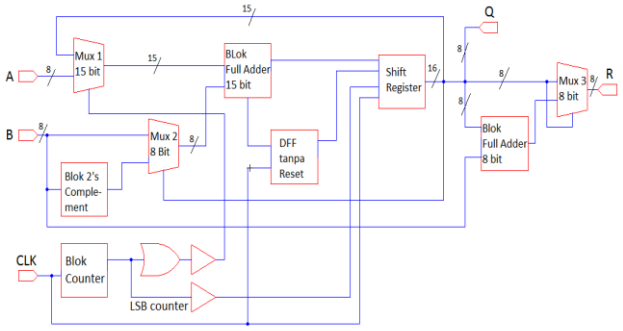
1. Blok 2's complement, berfungsi untuk mengubah divisor menjadi bilangan 2's complement-nya.
2. Blok full adder 15 bit, berfungsi untuk menjumlahkan partial remainder dengan bilangan 2's complement dari divisor.
3. Blok multiplexer, berfungsi untuk menyeleksi nilai mana yang akan dijumlahkan dengan bilangan 2's complement dari divisor (dividend atau partial remainder) dan untuk menyeleksi nilai partial remainder mana yang akan digeser. Pada divider ini,

dibutuhkan 2 buah blok multiplexer 2 ke 1 sebesar 15 bit.

4. Blok counter, berfungsi sebagai kendali masukan dari salah satu blok multiplexer dan blok shift registers. Counter yang diperlukan adalah counter yang dapat melakukan perhitungan mulai dari 0000_2 hingga 10000_2, karena 10000_2 merupakan jumlah sinyal clock yang dibutuhkan sebuah divider untuk menyelesaikan operasi pembagian.
5. Blok shift registers, berfungsi untuk mengatur kapan suatu shift registers melakukan proses load dan kapan melakukan shift.

2.3. Perancangan Divider

Apabila setiap unit-unit dasar dan blok-blok sudah sesuai secara fungsional, maka langkah selanjutnya adalah melakukan perancangan divider. Gambar 1 menunjukkan skematik dari divider yang dirancang.



Gambar 1. Skematik dari divider

Divider yang dirancang memiliki masukan berupa bilangan yang dibagi (dividend) sebesar 8 bit, pembagi (divisor) sebesar 8 bit, dan sinyal clock. Pada Gambar 1, dividend dinotasikan oleh A, divisor dinotasikan oleh B, dan sinyal clock dinotasikan oleh CLK. Untuk keluarannya, terdiri atas 16 bit bilangan yang terdiri atas hasil bagi (quotient) dan sisa bagi (remainder). Pada Gambar 1, keluaran dinotasikan oleh Out. Misal Out0 adalah least significant bit (LSB) dan Out16 adalah most significant bit (MSB), maka Out0 hingga Out7 merupakan quotient sementara Out9 hingga Out15 merupakan remainder.

Setelah perancangan tahap skematik selesai, langkah selanjutnya adalah perancangan pada tahap layout. Gambar 2 menunjukkan layout dari divider yang dirancang. A0 hingga A7 merupakan dividend sementara B0 hingga B7 merupakan divisor. Daerah 1 hingga daerah 5 merupakan blok-blok penyusun divider. Daerah 1 merupakan blok 2's complement, daerah 2 merupakan blok counter, daerah 3 merupakan blok full adder 15 bit, daerah 4 merupakan blok shift registers, dan daerah 5 merupakan salah satu dari blok multiplexer.

Gambar 2. Layout dari Divider

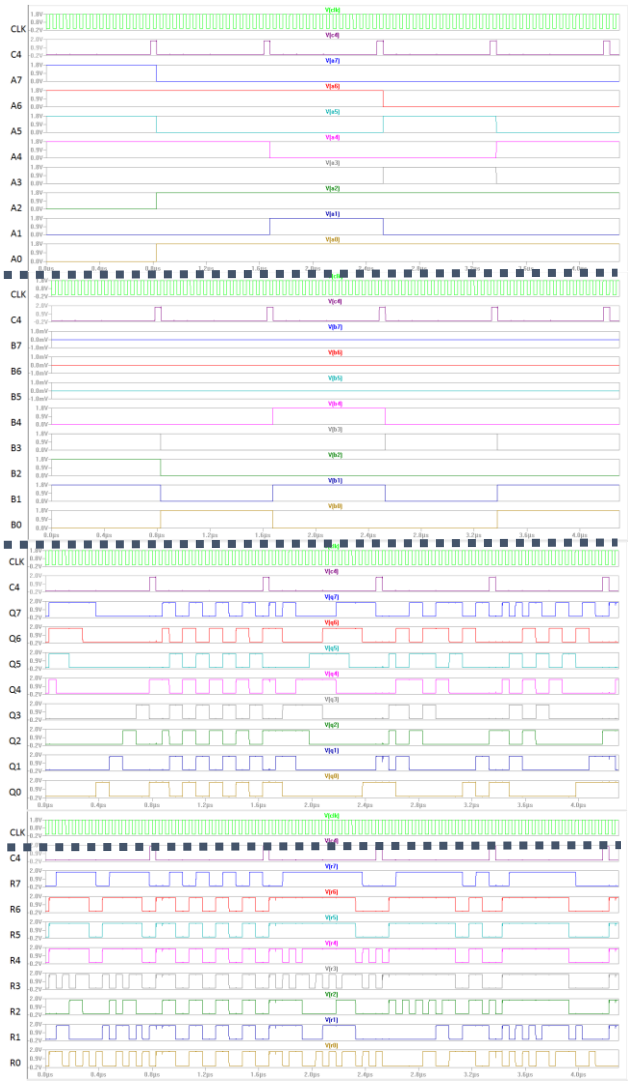
3. Hasil dan Analisa

Setelah perancangan layout dari divider selesai dilakukan, tahap selanjutnya adalah memberikan node masukan dan memberikan spice code pada layout. Setelah selesai, dilanjutkan dengan melakukan pengujian fungsional, penghitungan jumlah transistor, pengukuran luas, area coverage, dan timing delay. Untuk melakukan pengujian fungsional dan pengukuran timing delay, layout dari divider diekstrak menggunakan perangkat lunak LTspice.

3.1. Pengujian Fungsional

Karena divider yang dirancang memiliki besar 8 bit, maka terdapat 65.280 kemungkinan percobaan yang dapat dilakukan (dengan menganggap pembagain dengan bilangan 0 tidak terjadi). Namun pada pengujian fungsional ini, hanya dilakukan 40 percobaan dengan data yang berbeda (40 dividend dan 40 divisor). Hal ini dikarenakan 40 percobaan yang dilakukan sudah cukup memenuhi variasi percobaan yang diperlukan, misalnya seperti pembagian genap dengan ganjil, pembagian bilangan yang lebih kecil dengan yang lebih besar, pembagian yang menghasilkan hasil bagi bernilai besar namun sisa bagi bernilai kecil atau sebaliknya, dan variasi lainnya.

Empat puluh data yang diambil ini kemudian dibagi menjadi 8 tahap, sehingga setiap tahap terdiri atas 5 data. Gambar 3 menunjukkan gelombang keluaran dari salah satu tahap yang dilakukan, yaitu tahap 1.



Gambar 3. Pengujian divider tahap 1

Pada Gambar 3, CLK mewakili sinyal clock, C4 mewakili MSB dari counter, A0 hingga A7 mewakili dividend, B0 hingga B7 mewakili divisor, Q0 hingga Q7 mewakili quotient, dan R0 hingga R7 mewakili remainder. Perhatikan bahwa pada A, B, Q, dan R, angka 0 menunjukkan bahwa bilangan tersebut adalah LSB serta angka 7 menunjukkan bahwa bilangan tersebut adalah MSB.

Seperti yang telah dijelaskan pada subbab 2.2, divider yang dirancang membutuhkan sinyal clock sebanyak 100002 untuk menyelesaikan operasi yang dilakukan. Oleh karena itu, MSB dari counter berfungsi sebagai sinyal penanda untuk melihat apakah proses yang dilakukan sudah benar secara fungsional atau belum. Singkatnya, hasil keluaran (quotient dan remainder) dapat dilihat ketika C4 bernilai 1. Sementara ketika C4 bernilai 0, keluaran yang dihasilkan dapat diabaikan karena keluaran tersebut merupakan hasil pembagian yang belum selesai.

Sebagai contoh, pada tahap 1 ini dividend awal yang diberikan adalah 000101012 dan divisor akhir yang diberikan adalah 000000112. Ketika C4 bernilai 1, dapat dilihat bahwa keluaran quotient bernilai 000001112 dan keluaran remainder bernilai 000000002. Hal ini menunjukkan operasi yang dilakukan sudah sesuai secara fungional.

Tabel 1. menunjukkan hasil pengujian fungsional pada setiap tahap. Dari tabel 1 dapat dilihat bahwa divider yang dirancang telah sesuai secara fungsional.

Tabel 1. Pengujian fungsional divider.

No.	Masukan		Keluaran	
	Dividend	*Divisor*	*Quotient*	*Remainder*
1.	11110000	00001110	00010001	00000010
2.	01010101	00000001	01010101	00000000
3.	01000111	00010010	00000011	00010001
4.	00101101	00001000	00000101	00000101
5.	00010101	00000011	00000111	00000000
6.	11111111	00000010	01111111	00000001
7.	10101010	01111101	00000001	00101101
8.	11011100	00010011	00001011	00001011
9.	10010011	00000011	00110001	00000000
10.	10111110	00010001	00001011	00000011
11.	11111111	00000001	11111111	00000000
12.	01111000	01001001	00000001	00101111
13.	01100011	01100010	00000001	00000001
14.	11100111	00001001	00011001	00000110
15.	01011001	00000101	00010001	00000100
16.	10001001	00000111	00010011	00000100
17.	00001111	00001010	00000001	00000101
18.	11101000	00000111	00100001	00000001
19.	11111010	00110010	00000101	00000001
20.	10101001	00001111	00001011	00000100
21.	10010101	00000111	00010101	00000010
22.	11111010	00011010	00001001	00010000
23.	11001000	00001000	00011001	00000000
24.	11000110	00000100	00110001	00000010
25.	10110101	01101111	00000001	01000110
26.	00011001	00001000	00000011	00000001
27.	10011101	00001010	00001111	00000111
28.	00010010	00000110	00000011	00000000
29.	10100101	00000101	00100001	00000000
30.	10111010	01100101	00000001	01010101
31.	11001010	00000110	00100001	00000100
32.	10101000	00001000	00010101	00000000
33.	10100110	00001011	00001111	00000101
34.	01011011	01011011	00000001	00000000
35.	10011000	00010100	00000111	00001100
36.	00101111	00001001	00000101	00000010
37.	00100001	00000011	00001011	00000000
38.	11111001	01001101	00000011	00010010
39.	01110000	00111011	00000001	00110101
40.	01100100	01000011	00000001	00100001

3.2. Pengukuran Luas dan Area Coverage

Untuk penghitungan luas dan area coverage dari divider yang dirancang, dapat digunakan fitur 'Check Area Coverage' pada perangkat lunak Electric. Dari hasil pengukuran, didapati bahwa divider yang dirancang memiliki dimensi sebesar 4.891.122,5 λ2. Karena teknologi yang digunakan adalah 180nm, maka besarnya λ

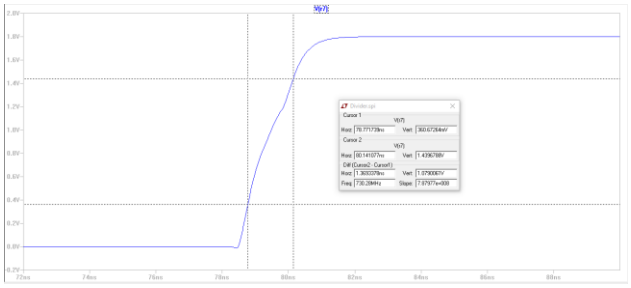
adalah 90nm. Dengan demikian, luas dari divider yang dirancang adalah sebesar 0,04mm2.

Setelah pengukuran luas, langkah selanjutnya adalah menghitung area coverage. Area coverage dapat diartikan sebagai seberapa rapat daerah yang digunakan dalam perancangan. Semakin tinggi nilai area coverage-nya, maka rancangan tersebut semakin baik karena susunannya semakin rapat sehingga tidak ada daerah yang kosong atau terbuang. Pada perancangan divider ini, didapati besar area coverage-nya adalah 40,115%.

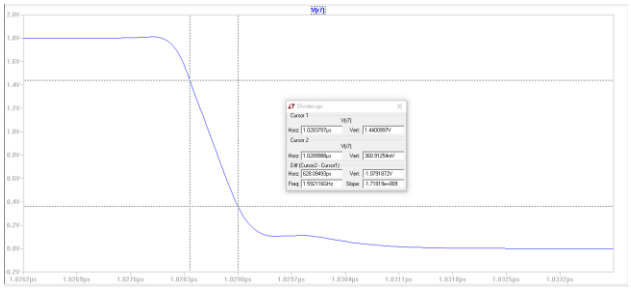
3.3. Pengukuran Timing Delay

Pengukuran timing delay dilakukan pada gelombang keluaran hasil akhir pada setiap variasi masukan di setiap tahap. Timing delay yang diukur meliputi rise time (Tr), fall time (Tf), propagation delay rise time (Tpdr), propagation delay fall time (Tpdf), dan propagation delay time (Tpd).

Tabel 2 menunjukkan pengukuran propagation delay time yang dilakukan. Dari Tabel 2, dapat dilihat bahwa propagation delay time terbesar adalah 3,938ns (masukan nomor 29). Setiap data yang diambil diukur pada seluruh variasi masukan di setiap tahap dengan mencari keluaran mana yang memiliki delay paling besar. Gambar 4 dan Gambar 5 beruturut.-turut menunjukkan pengukuran Tr dan Tf terbesar, sementara Gambar 6 dan Gambar 7 berturut-turut menunjukkan pengukuran Tpd dan Tpdf yang menghasilkan nilai Tpd dengan nilai terbesar.



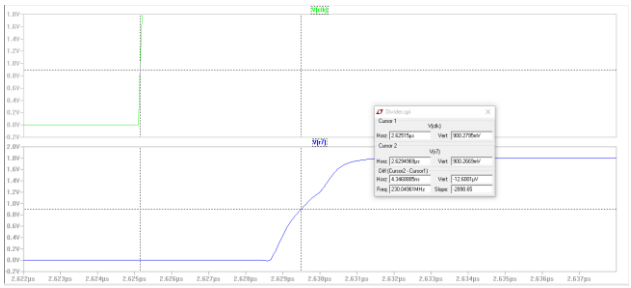
Gambar 5. Pengukuran tr divider



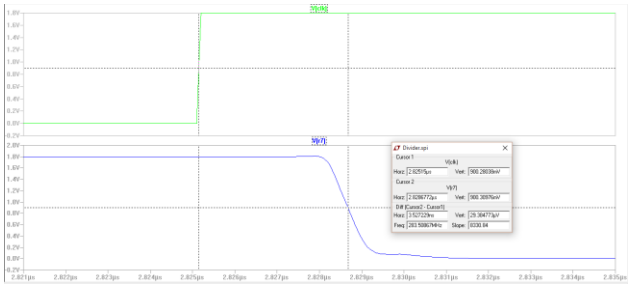
Gambar 6. Pengukuran tf divider

Tabel 2. Pengukuran time propagation delay

No.	Masukan		Tpd
	Dividend	*Divisor*	(ns)
1.	11110000	00001110	3,848
2.	01010101	00000001	3,935
3.	01000111	00010010	3,884
4.	00101101	00001000	3,871
5.	00010101	00000011	3,929
6.	11111111	00000010	3,821
7.	10101010	01111101	2,874
8.	11011100	00010011	3,867
9.	10010011	00000011	3,913
10.	10111110	00010001	3,934
11.	11111111	00000001	2,590
12.	01111000	01001001	2,898
13.	01100011	01100010	2,883
14.	11100111	00001001	3,912
15.	01011001	00000101	3,914
16.	10001001	00000111	3,894
17.	00001111	00001010	3,883
18.	11101000	00000111	3,902
19.	11111010	00110010	3,834
20.	10101001	00001111	3,873
21.	10010101	00000111	3,906
22.	11111010	00011010	3,838
23.	11001000	00001000	3,875
24.	11000110	00000100	3,877
25.	10110101	01101111	2,859
26.	00011001	00001000	3,876
27.	10011101	00001010	3,874
28.	00010010	00000110	3,751
29.	10100101	00000101	3,938
30.	10111010	01100101	2,894
31.	11001010	00000110	3,890
32.	10101000	00001000	3,885
33.	10100110	00001011	3,880
34.	1011011	01011011	2,867
35.	10011000	00010100	3,874
36.	00101111	00001001	3,896
37.	00100001	00000011	3,920
38.	11111001	01001101	2,903
39.	01110000	00111011	3,818
40.	01100100	01000011	2,914



Gambar 6. Pengukuran tpdr dari divider



Gambar 7. Pengukuran tpdf dari divider

Dari hasil pengukuran dengan nilai terbesar yang dilakukan, didapatkan Tr sebesar 1,37ns, Tf sebesar 0,628ns, Tpdr sebesar 4,349ns, dan Tpdf sebesar 3,528ns. Tabel 3 merangkum hasil pengukuran timing delay maksimal yang dilakukan pada divider.

Tabel 3. Pengukuran timing delay maksimal

Tr (ns)	Tf (ns)	Tpdr (ns)	Tpdf (ns)	Tpd (ns)
1,37	0,628	4,349	3,528	3,938

3.4. Analisa

Pada subbab 2.3 telah dijelaskan bahwa selain dividend dan divisor, divider yang dirancang memiliki masukan sinyal clock. Besarnya sinyal clock sangat penting dalam rancangan sekuensial. Apabila sinyal clock yang diberikan terlalu kecil periodenya, maka ada kemungkinan nilai yang diinginkan menjadi tidak sesuai. Apabila sinyal clock yang diberikan terlalu besar periodenya, maka waktu yang dibutuhkan untuk menyelesaikan suatu proses menjadi lebih lama. Untuk itu, diperlukan sinyal clock dengan periode atau frekuensi yang aman agar proses yang dilakukan menghasilkan keluaran yang sesuai, namun juga tidak memakan waktu yang lama.

Pada pengujian yang dilakukan, sinyal clock yang digunakan adalah sebesar 50ns, 16ns, 14ns, 12ns, dan 10ns. Dari keempat sinyal clock tersebut, percobaan dengan sinyal clock 12ns dan 10ns terdapat kesalahan fungsional. Hal ini dikarenakan pada saat sinyal clock sebesar 12ns dan 10ns, blok full-adder yang digunakan belum selesai dalam melakukan proses penjumlahan sehingga nilai keluaran full-adder yang digeser ke shift register adalah nilai penjumlahan yang belum selesai. Hal ini menyebabkan hasil pembagian menjadi tidak fungsional.

Dengan menggunakan sinyal clock sebesar 50ns, maka proses yang dibutuhkan oleh suati sinyal clock yang dibutuhkan oleh divider ini untuk menyelesaikan perhitungan pada satu siklus adalah sebesar 50ns x 16, yaitu sekitar 800ns. Enam belas merupakan banyaknya sinyal clock yang dibutuhkan untuk menyelesaikan pembagian dalam satu siklus.

4. Kesimpulan

Dari pengujian yang telah dilakukan, divider yang dirancang pada penelitian ini telah sesuai secara fungsional. Divider yang dirancang memiliki luas sebesar 0,04mm2, area coverage sebesar 40,115%, dan propagation delay sebesar 3,938ns. Divider ini diuji dengan sinyal clock berfrekuensi 20MHz. Untuk pengembangan selanjutnya, dapat dirancang suatu divider dengan teknologi yang berbeda, sehingga menghasilkan luas dan propagation delay time yang berbeda untuk dibandingkan lebih jauh seperti apa perbedaannya.

Referensi

[1]. P. Nair, D. Kudithipudi, and E. John, “Design and Implementation of a CMOS Non-Restoring Divider,” pp. 211–217, 2006.

[2]. S. K. Panda, “A Novel Vedic Divider Architecture with Reduced Delay for VLSI Applications,” vol. 120, no. 17, pp. 31–36, 2015.

[3]. C.-L. Wey, “Design of A Fast Radix-4 SRT Divider and its VLSI Implementation,” vol. 146, pp. 205–210, 1999.

[4]. A. S. Kharade, S. V Kapadiya, and R. Chavan, “4-Bit Arithmetic and Logic Unit Design Using Sturctural Modelling in VHDL,” vol. 2, no. 3, pp. 1–10, 2013.

[5]. J. Melorose, R. Perroy, and S. Careas, “Division,” Statew. Agric. L. Use Baseline 2015, vol. 1, 2015.

[6]. M. M. Mano and C. Kime, Logic and Computer Desing Fundamentals. 2014.

[7]. R. J. Tocci, Digital Systems: principles and applications. Pearson Education India, 1980.

[8]. N. E. H. Weste and D. M. Harris, CMOS VLSI Design: A Circuits and Systems Perspective, vol. 53, no. 9. 2013.

[9]. J. F. Wakerly, Digital Design Practices and Principles Third Edition. .

[10]. “Chapter 2 MOS Transistors,” vol. 1, pp. 1–22.

[11]. Mosis, “Scalable CMOS Design Rules”, 2009.