

# DESAIN *TIMING* GENERATOR SEDERHANA PADA CCD LINIER MENGUNAKAN MODUL FPGA

## *SIMPLE TIMING GENERATOR DESIGN FOR LINEAR CCD USING FPGA MODULE*

A. Hadi Syafrudin

Pusat Teknologi Satelit - Lembaga Penerbangan dan Antariksa Nasional (LAPAN)

ahaddi@yahoo.com

### **Abstrak**

CCD linier secara umum digunakan sebagai sensor *imager* untuk satelit *remote sensing*. Pembuatan *imager* dari CCD membutuhkan komponen yang cepat dan berjalan paralel. Salah satu komponen yang sering digunakan adalah FPGA yang berfungsi mensinkronisasi berbagai komponen pendukung sampai dikeluarkannya sinyal video analog dan digitasi dengan *analog digital converter*. Proses sinkronisasi ini umumnya berupa *timing generator* yang spesifik dirancang sesuai dengan komponen yang digunakan. Untuk menyederhanakan pemrograman FPGA maka digunakan analisa *timing* diagram dari masing masing komponen dan dibuat tabel keluaran pada tiap fase konter FPGA. Cara ini sederhana dari segi pemrograman tetapi diperlukan analisa manual untuk membuat tabel keluaran dari *timing generator*. Pada penelitian ini akan dibahas proses desain *timing* generator, tabel keluaran sinyal per fase konter, hasil berupa sinyal analog dan hasil gambar dari CCD linier. Dengan cara sederhana tersebut telah berhasil dibuat *timing* generator yang handal untuk menghasilkan gambar. Kata kunci: *Timing Generator*, CCD Linier ,FPGA.

### **Abstract**

*Linear CCD generally used for imager sensor in remote sensing Satellite. For build imager from CCD sensor need high speed device and parallel processing. Ones frequently device to do this job is FPGA which function to synchronize some support component until output of video analogue signal and digitation with analogue digital converter. Synchronization process generally called timing generator which specially design for some component used. For simplification of FPGA so timing diagram have analyzed from every component and output table created on every counter FPGA phase. This way is simple for programmer but need manual analysis for create output table for timing generator. In this research will discuss about design process of timing generator, output signal table each counter phase, analogue signal result and digital image result from linear CCD. This simple way is success to create robust timing generator for make digital images.*

*Keywords: Timing Generator, linear CCD, FPGA.*

## **1. PENDAHULUAN**

CCD linier secara umum digunakan sebagai sensor *imager* untuk satelit *remote sensing*. CCD linier banyak digunakan di industri pada aplikasi *machine vision* dan satelit dengan muatan optik. CCD digunakan pada *onboard* satelit dikarenakan *noise* yang kecil dan dinamik *range* yang tinggi dibandingkan sensor CMOS [1]. Selain keunggulannya CCD memerlukan komponen tambahan semisal *Analog digital converter* (ADC) dengan *Correlated Double Sampling* (CDS) [2] dan *pin driver* untuk memperkuat sinyal *control* dan *trigger*.

Salah satu komponen yang sering digunakan adalah FPGA yang berfungsi mensinkronisasi berbagai komponen pendukung sampai dikeluarkannya sinyal video analog dan digitasi dengan *ADC*. Proses sinkronisasi ini umumnya berupa *timing* generator yang spesifik dirancang sesuai dengan komponen yang digunakan. Pembuatan *timing* generator merupakan salah satu kunci utama dalam pembuatan kamera. Untuk membuat *timing* generator yang dapat diimplementasikan terkadang butuh waktu beberapa tahun [3].

Sirkuit *driver* CCD tradisional sangat sulit untuk memenuhi aplikasi perangkat CCD berkecepatan tinggi karena kompleksitas dan kemampuan anti-interferensi yang buruk[4]. Hal ini disebabkan output

dan transmisi sinyal berupa analog sehingga apabila desain PCB dimana sinyal digital dan analog berdekatan menyebabkan interferensi. Dalam penelitian ini perlu dilakukan pembahasan berbeda antara desain *timing* generator dan *noise isolation* CCD. Ini perlu diketahui diawal dikarenakan pada makalah ini fokus membahas tentang FPGA sebagai generator.

FPGA telah menunjukkan kinerja yang sangat tinggi terlepas dari frekuensi operasional rendah mereka dengan sepenuhnya mengekstraksi paralelisme[5]. Pemrograman paralelisme FPGA menyebabkan komponen ini dapat berjalan lebih cepat dikarenakan untuk antarmuka pin yang jumlahnya puluhan sampai ratusan dapat berjalan dengan arah input output dan fase yang berbeda. Walaupun bekerja dengan kecepatan yang bervariasi pada dasarnya system di dalam tetap menggunakan *clock* yang sama. *Clock* yang sama ini bisa disebut dengan *clock* utama. *Clock* utama ini mempunyai frekuensi yang jauh lebih tinggi dari seluruh antarmuka yang digunakan. Perbedaan tersebut hanya berupa konter *clock* dengan jumlah tertentu sehingga menjamin semua sistem dapat berjalan sinkron.

*Synchronous interface controller* dapat dilakukan ketika spesifikasi komunikasi didasarkan pada *timing* diagram [6]. Secara tradisional, *timing behaviour* pada sirkuit digital telah dijelaskan dengan menggunakan *timing diagram* atau *waveform* [7]. *Timing* diagram secara umum mempunyai nilai antara yang dapat didekati dengan nilai *clock* utama yang ada. Dengan menganalisa *timing* diagram tiap komponen dan menentukan jumlah *clock* serta *initial condition* maka dapat ditentukan nilai *timing* generator output yang dikeluarkan pada setiap *clock* utama bekerja.

Nilai output pada tiap *clock* utama dapat dianalisa dan didekati kemudian dibuat tabel antara konter dan nilai output. FPGA komersial yang tersedia secara luas didasarkan pada tabel *lookup* (LUTs) yang terdiri dari *array* SRAM dan multiplexer [8]. Benar tidaknya tabel yang telah dibuat dapat dianalisa dengan sinyal yang dikeluarkan CCD dan output seharusnya secara teori. Output CCD yang dihasilkan nantinya berupa sinyal video analog yang mempresentasikan nilai intensitas tiap piksel CCD.

Pada penelitian kali ini bertujuan mendesain dan mengimplementasikan *timing* generator CCD linier dengan menganalisa *timing* diagram dan membuat tabel antara output generator dan *clock* konter. *Timing* generator tersebut akan menghasilkan sinyal video analog dan gambar digital dengan menggunakan USB *module* yang ada. *Timing* generator akan mensinkronisasi antara CCD linier, ADC dengan CDS, FPGA Data Buffer, dan modul USB.

## 2. DESAIN *TIMING* GENERATOR MENGGUNAKAN FPGA

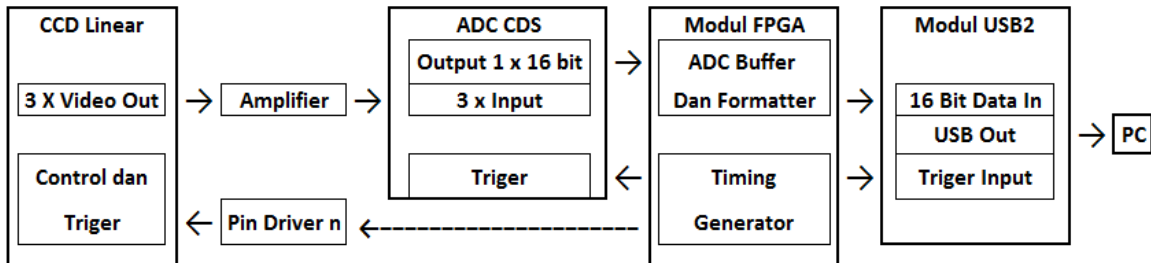
Pada Penelitian ini komponen utama yang digunakan adalah CCD linier dengan tipe KLI-8023[9], Modul FPGA dengan tipe Altera Cyclone IV yang ada pada modul terasic De0 nano[10], dan ADC 16 bit dengan CDS dengan tipe AD-9826[11]. Selain dari tiga komponen tersebut ada beberapa komponen pendukung yang dapat dilihat pada *datasheet* tiga komponen utama atau yang mempunyai fungsi yang sama dengan komponen pendukung tersebut. Pengukuran sinyal video output digunakan osiloskop. Sedangkan perekaman data ADC digital digunakan modul USB cypress FX2 beserta *software* perekam dan diterjemahkan dengan *software* untuk pengelompokan warna.

Dalam penelitian ini dilakukan dengan beberapa tahap perencanaan desain, implementasi, pengujian dan analisa. Detail metodologi kegiatan ini adalah sebagai berikut:

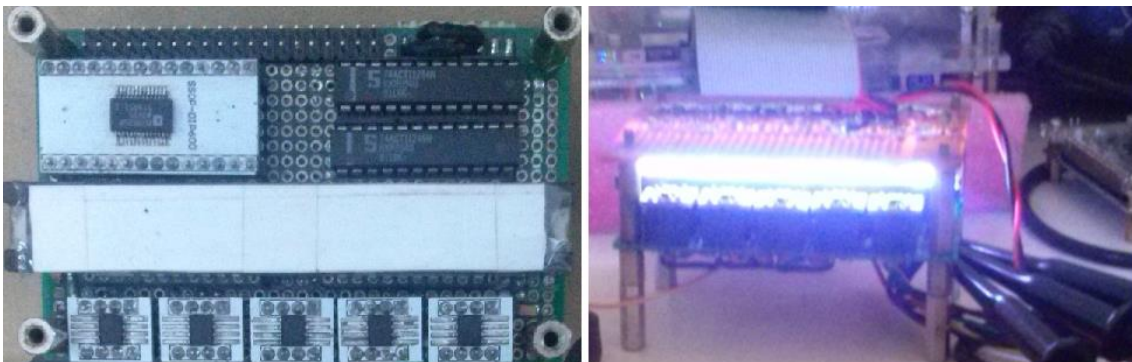
- Pembuatan desain *hardware* dan *interface* antara FPGA, CCD, ADC, modul USB, dan pin *driver* dan komponen elektronika aktif pasif sesuai *datasheet* komponen.
- Mempelajari prinsip kerja CCD linier, ADC, dan modul USB yang digunakan untuk dikelompokkan menjadi beberapa simbol diagram FPGA. Pada masing masing simbol akan dibuat *timing* generator antara tabel output dan konter *clock* serta *interface* antar simbol. Pemrograman akan dilakukan dengan dua metode, yaitu simbol dan vhdl. Di mana step ini mendesain simbol yang digunakan serta membuat koneksi antar simbol.
- Pembuatan vhdl dan tabel output untuk setiap simbol yang telah dibuat berdasarkan analisa *timing* diagram tiap komponen.
- Pengujian sinyal output analog dengan osiloskop dengan mengubah intensitas sinyal masukan serta membandingkan dengan sinyal ideal dari *datasheet*.
- Pengujian perekaman data digital dengan usb dengan memberikan halangan pada sumber cahaya input untuk meyakinkan bahwa CCD linier dapat bekerja dengan baik.

### 2.1. Desain Hardware dan Interface

Desain *hardware* yang telah dibuat ditunjukkan pada Gambar 1. Modul FPGA mempunyai fungsi utama sebagai *timing* generator dan ADC Buffer. Pada Gambar 2 merupakan tampilan alat yang telah dibuat. CCD linier ditutupi kertas putih tipis yang berlaku sebagai atenuator cahaya. Dengan membuat lampu led berbaris sebagai cahaya input yang dapat diatur intensitasnya.



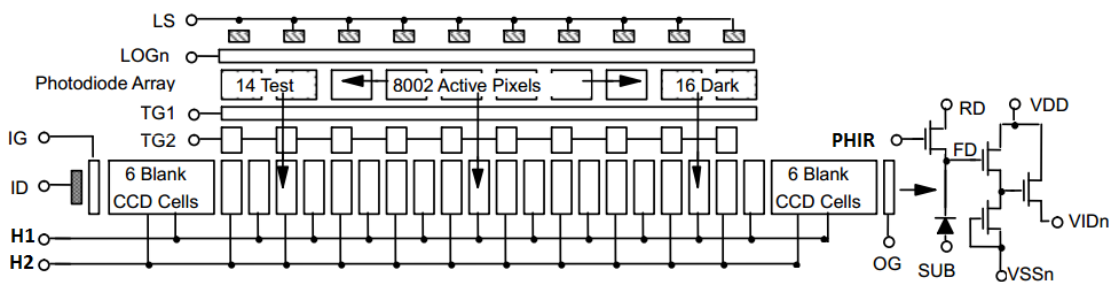
Gambar 1. Diagram blok sistem yang terdiri dari CCD linier, amplifier dan pin driver, ADC CDS 16 bit, Timing Generator pada Modul FPGA, dan Modul USB.



Gambar 2. Modul CCD linier, pin driver dan amplifier serta pengujian dengan LED yang dipasang pada permukaan modul CCD.

### 2.2. Prinsip Kerja CCD Linier dan Simbol VHDL

Prinsip kerja CCD ditunjukkan pada Gambar 3 dimana terdapat dua proses yaitu pengambilan dan pengiriman Data. Pengambilan data proses konversi cahaya ketegangan sedangkan pengiriman data proses keluarnya sinyal video berupa tegangan secara serial tiap kanal.

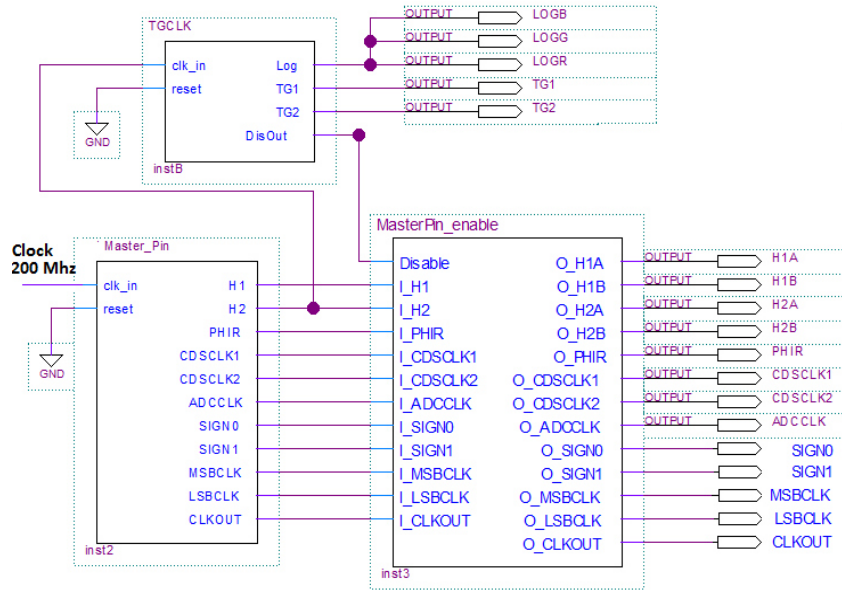


Gambar 3. Skema CCD KLI-8023 [9]

Proses pengambilan data dimulai dari aktifnya *exposure* (Logn) yang memungkinkan cahaya masuk ke *photodiode array* dan disimpan pada *Transfer Gate 1* (TG1) dan TG2. Ketika terjadi *clocking* pada kedua sinyal tersebut maka data akan diteruskan pada jalur transmisi. Jalur transmisi akan mengeluarkan sinyal video seiring dengan *clock* H1 dan H2 secara *shifting* atau PISO(paralel input Serial output). Sinyal tersebut akan disinkronkan dengan *control* ADC dan tanda antara sinyal dan MSB LSB Output.

Pada Gambar 4 merupakan hasil desain Simbol FPGA dengan 3 blok. Ada dua simbol utama yaitu TGCLK dan Master\_pin serta on/off dari *master pin* dengan simbol *master\_pin\_enable*. TGCLK simbol akan aktif dengan *clock* input H2 sebesar 5 Mhz dan akan menonaktifkan *master\_pin\_enable* selama

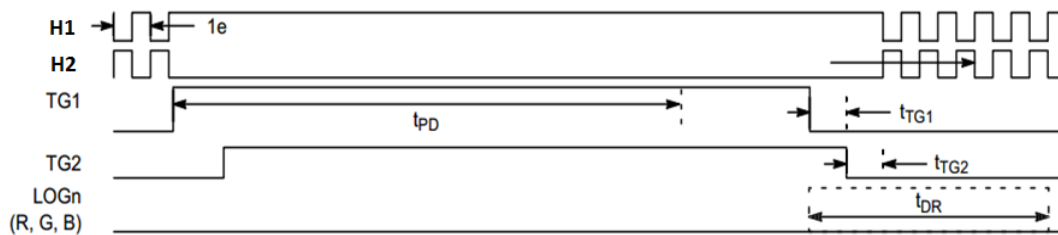
proses transisi Pengambilan gambar berlangsung. Sedangkan Master\_pin akan berjalan secara terus menerus.



Gambar 4. Simbol FPGA - Timing Generator CCD Linear (TGCLK Blok akan menghentikan jalur transmisi Sinyal Master\_Pin Setiap exposure/log mengambil data).

2.3. Desain Timing Generator Pengambilan Data

Modul FPGA mempunyai internal clock sebesar 50 Mhz dengan PLL 4 kali maka dapat diperoleh clock utama sebesar 200Mhz atau 5 ns[10]. H2 merupakan 40 konter adari clock utama yang digunakan untuk menghasilkan sinyal pada transmisi data. Pada desain ditentukan melakukan pengambilan gambar selama 2.7mS atau 13500 clock dan exposure time 2.5 mS atau 12500 clock dengan satu clock H2 sebesar 200nS atau 5 Mhz. Dari gambar clock H2 adalah senilai 1e. pada saat TG1 dan TG2 aktif yang merupakan peralihan pengambilan data dan pengiriman data clock H1 dan H2 dimatikan. Hal ini dijelaskan dengan timing diagram Gambar 5.



Gambar 5. Timing diagram pengambilan Data [9]

Dari hasil timing diagram pada Gambar 5 dapat dihasilkan tabel output seperti pada Tabel 1 dimana periode sebesar 0-13499 atau 13500 clock seperti pada desain. Dari tabel pin output akan mengeluarkan data yang sama sebelum batas konter terpenuhi. Implementasi pada bahasa VHDL pada Tabel 1 ditunjukkan pada Gambar 6.

Tabel 1. Tabel Output Pin Pengambilan data TGCLK

Parameter	= 13499	< 18	< 21	< 24	< 1018	< 8068	< 13499
Log	1	1	0	0	0	1	1
TG1	0	1	0	0	0	0	0
TG2	0	1	1	0	0	0	0
DisOut	1	1	1	1	0	0	1

```

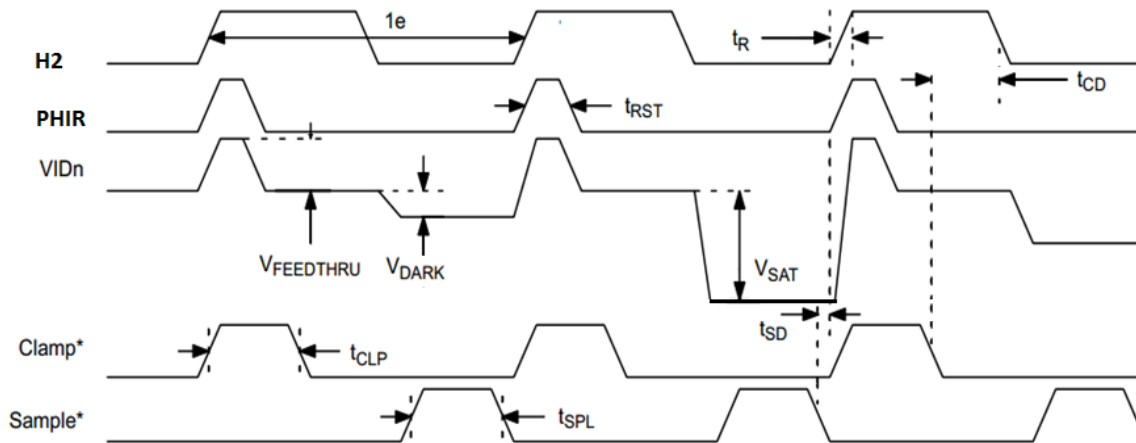
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity TGCLK is
5  Port (
6      clk_in : in  STD_LOGIC;
7      reset  : in  STD_LOGIC;
8      Log    : out STD_LOGIC;
9      TG1    : out STD_LOGIC;
10     TG2    : out STD_LOGIC;
11     DisOut : out STD_LOGIC
12 );
13 end TGCLK;
14
15 architecture Behavioral of TGCLK is
16     signal counter : integer range 0 to 13499 := 0;
17     begin
18     process (reset, clk_in) begin
19         if (reset = '1') then
20             counter <= 0;
21         elsif rising_edge(clk_in) then
22             if (counter = 13499) then
23                 Log <= '1';
24                 TG1 <= '0';
25                 TG2 <= '0';
26                 DisOut <= '1';
27                 counter <= 0;
28             elsif ( counter < 18) then
29
30             elsif ( counter < 21) then
31
32             elsif ( counter < 24) then
33
34             elsif ( counter < 1018) then
35
36             elsif ( counter < 8068) then
37
38             elsif ( counter < 13499) then
39
40             end if;
41         end process;
42     end Behavioral;

```

Gambar 6. Bahasa VHDL pada simbol TGCLK

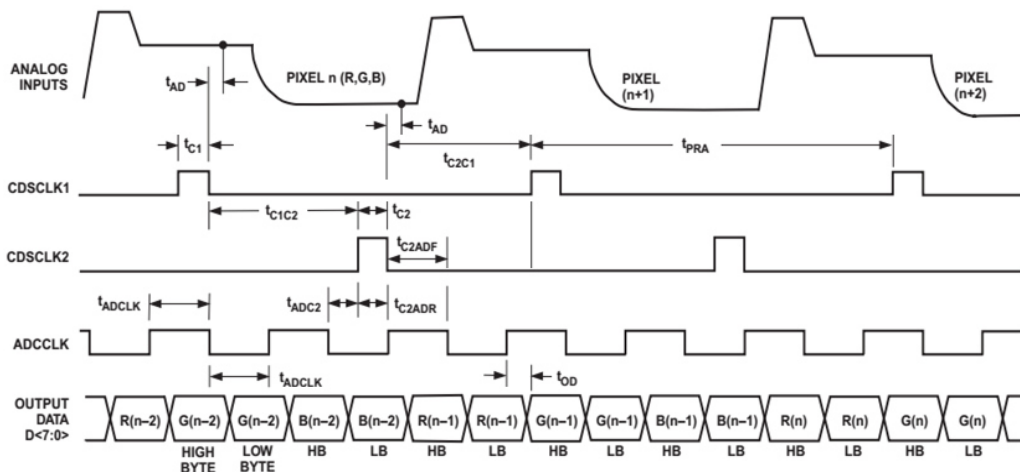
2.4. Desain Timing Generator Pengiriman Data

Untuk membuat tabel output pada pengiriman data harus dianalisa antara *timing* CCD dan ADC terutama pada saat melakukan CDS yaitu saat clamp dan sample. Di mana intensitas piksel diwakili dengan jarak tegangan antara dua titik tersebut. Sinyal PHIR akan me-reset transistor pada Gambar 3 dan mempunyai kesamaan dengan *peak* sinyal video output.



Gambar 7. Timing diagram transmisi data CCD KLI-8023 [9]

Dari analisa *timing* diagram KLI-8023 dan AD-9826 diperoleh Tabel 2. Di mana pada tabel tersebut mempunyai periode 40 *clock* atau 5 Mhz dengan *clock* terkecil 5 nS atau 200 Mhz. Mode yang digunakan ADC ada 3 mode CDS dengan 3 input dan output 16 bit bergantian antara out RGB dan MSB-LSB. ADC *clock* mempunyai kecepatan sampling 15 Mhz ( 5 Mhz untuk tiap kanal).



Gambar 8. Timing diagram 3 mode ADC9826 [11]

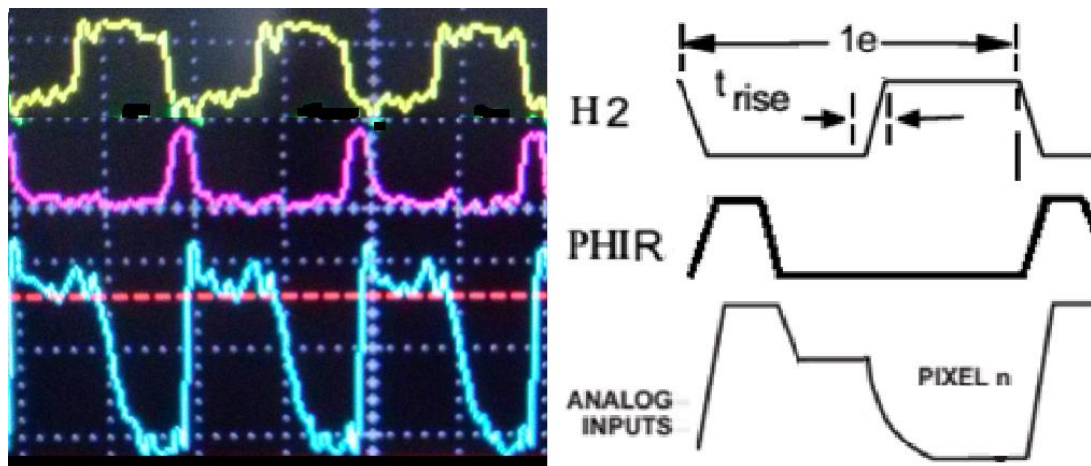
**Tabel 2. Tabel output pin – Konter clock pengiriman data, 1 clock utama mewakili 5 nS (200 Mhz) dan 1 clock CCD diwakili 40 clock utama (5 Mhz).**

	master pin (1 clk = 5 ns = 200Mhz)																																										
	39	<1	<3	<4	<5	<8	<10	<11	<12	<13	<15	<16	<17	<18	<20	<23	<24	<25	<26	<28	<29	<30	<31	<33	<36	<37	<38	<39	39														
H1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1						
H2	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
PHIR	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1					
CDSCLK1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
CDSCLK2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0		
ADCCLK	1	1	1	1	0	0	0	0	1	1	1	1	1	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	
SIGN0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0		
SIGN1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
MSBCLK	0	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
LSBCLK	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	
CLKOUT	1	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

### 3. HASIL DAN PEMBAHASAN

#### 3.1. Pengujian Output Sinyal Analog

Pengukuran sinyal output dilakukan dengan osiloskop dan difokuskan pada sinyal H2 phir dan video output dan dibandingkan dengan sinyal ideal video output. Dari perbandingan sinyal terdapat *noise* dari sinyal output tetapi bentuk secara umum masih mewakili sinyal yang sama. Hasil pengujian ditampilkan pada Gambar 9.

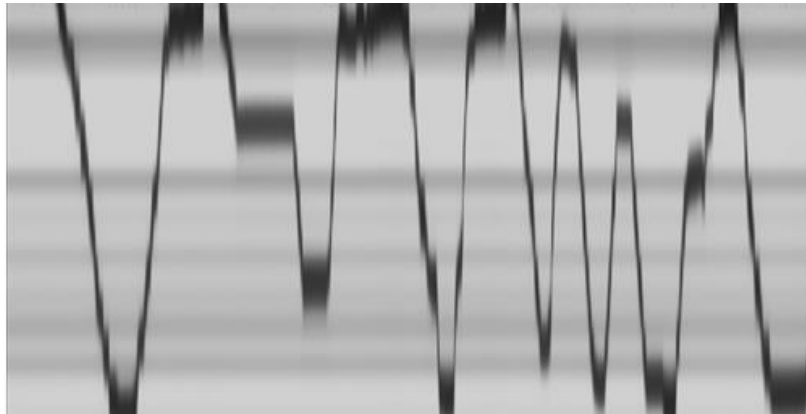


**Gambar 9. Hasil output video dibandingkan dengan hasil ideal dari data sheet (Tampak ada *noise* atau *jitter* pada sinyal yang ditampilkan osiloskop)**

Penyebab *jitter* atau *noise* secara bisa disebabkan oleh *clock* FPGA itu sendiri, Peletakan komponen di PCB atau *noise* dari *power supply* yang digunakan. Model *noise* CCD adalah kombinasi dari campuran *noise fixed-pattern* dan *multiplicative Gaussian noise* dan campuran *noise signal-independent* [12]. *Switching current* yang ditarik oleh *integrated circuit* (IC) menciptakan *noise power supply* yang dinamis pada IC dan pada *printed circuit board* (PCB), yang pada gilirannya menyebabkan *jitter* pada sinyal input output dan mengurangi frekuensi *clock* maksimum [13]. Aplikasi berkecepatan tinggi menimbulkan spektrum *noise* yang tinggi dan membuat disain sistem menjadi rumit [14].

#### 3.2. Pengujian pengambilan Gambar Digital

Dari hasil pengujian dengan obyek bergerak menggunakan modul usb didapat hasil seperti Gambar 10. Dari gambar terlihat *pattern* yang sama dari kanan ke kiri diakibatkan ketidakseragaman detektor atau ketidakseragaman cahaya yang masuk ke sensor. Dari gambar penghalang tampak sensor dapat berfungsi dengan baik dengan melihat *pattern* bayangan yang ada. Dengan sensor linier dengan obyek yang diam akan menghasilkan *frame* dengan *pattern* yang sama berkebalikan dengan obyek yang bergerak.



**Gambar 10.** Hasil *scanning* CCD linier dengan penghalang yang digerakkan dari atas bawah (Arah *scanning* dari kiri ke kanan)

#### 4. KESIMPULAN

Desain *timing* generator dengan menggunakan FPGA telah berhasil dilakukan dengan menghasilkan sinyal analog dengan sedikit *noise* dan gambar *full scanning* dengan *interface* usb. Analisa *timing* diagram tiap komponen dapat menghasilkan tabel konter *clock* dan output tabel yang dengan mudah diimplementasikan pada FPGA. Perlu adanya kegiatan penelitian lanjutan untuk mengetahui pasti sumber *noise* dan memperbaiki penelitian yang telah dilakukan. Perlu dilakukan perbandingan akibat *noise* jika digunakan baterai sebagai *power supply*-nya.

#### UCAPAN TERIMA KASIH

Dengan terselesaikannya karya ilmiah ini penulis mengucapkan banyak terimakasih kepada teman teman sesama peneliti dan perekayasa di Pusat Teknologi Satelit atas saran, pemberian semangat, motivasi dan diskusi yang membangun. Para ketua Management dan tim pendukung yang memberikan tugas serta dukungan dana maupun tenaga serta kerja sama dalam melaksanakan kegiatan penelitian di Pusteksat LAPAN.

#### PERNYATAAN PENULIS

Penulis dengan ini menyatakan bahwa seluruh isi menjadi tanggung jawab penulis.

#### DAFTAR PUSTAKA

- [1] P. Shah, B. Soni, M. Waris, R. Kumaran, S. Mehta and A. R. Chowdhury, "Generic and programmable *Timing* Generator for CCD detectors," 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), New Delhi, 2014, pp. 1845-1851.
- [2] Z. Yuan, "Notice of Retraction Design of the time driving and image signal processing for linear CCD based on FPGA," 2011 International Conference on Consumer Electronics, Communications and Networks (CECNet), XianNing, 2011, pp. 3088-3090
- [3] Binhua Li ; Qian Song ; Chun He ; Jianhui Jin and Lin He. "Method to implement the CCD *timing* generator based on FPGA", Proc. SPIE 7742, High Energy, Optical, and Infrared Detectors for Astronomy IV, 77421Y (July 19, 2010);
- [4] T. Xu, X. Xiong and M. Li, "Design of FPGA based on linear *array* CCD *driver* circuit," 2009 9th International Conference on Electronic Measurement & Instruments, Beijing, 2009, pp. 2-766-2-769.

- [5] S. Asano, T. Maruyama and Y. Yamaguchi, "Performance comparison of FPGA, GPU and CPU in image processing," 2009 International Conference on Field Programmable Logic and Applications, Prague, 2009, pp. 126-131
- [6] El-Aboudi, E. M. Aboulhamid and E. Cerny, "Verificatiom of synchronous realizability of *interfaces* from *timing* diagram specifications," Proceedings of the Tenth International Conference on Microelectronics (Cat. No.98EX186), Monastir, 1998, pp. 103-106.
- [7] G. Borriello, "Formalized *timing* diagrams," [1992] Proceedings The European Conference on Design Automation, Brussels, 1992, pp. 372-377
- [8] P. S. Nair, S. Koppa and E. B. John, "A comparative analysis of coarse-grain and fine-grain power gating for FPGA *lookup* tables," 2009 52nd IEEE International Midwest Symposium on Circuits and Systems, Cancun, 2009, pp. 507-510.
- [9] [www.onsemi.com/pub/Collateral/KLI-8023-D.PDF](http://www.onsemi.com/pub/Collateral/KLI-8023-D.PDF)
- [10] [https://www.terasic.com.tw/attachment/archive/941/DE0-Nano-SoC\\_User\\_manual.pdf](https://www.terasic.com.tw/attachment/archive/941/DE0-Nano-SoC_User_manual.pdf)
- [11] <http://www.analog.com/media/en/technical-documentation/data-sheets/AD9826.pdf>
- [12] Y. Tian and Z. Wang, "A Method of CCD *Noise* Removal Using the Contourlet Transform in Digital Images," 2011 Fourth International Conference on Intelligent Computation Technology and Automation, Shenzhen, Guangdong, 2011, pp. 350-353
- [13] L. Ren et al., "Prediction of Power Supply *Noise* From Switching Activity in an FPGA," in IEEE Transactions on Electromagnetic Compatibility, vol. 56, no. 3, pp. 699-706, June 2014.
- [14] Lian Nee Soh, Hui Lee Teng, P. Boyle, Man On Wong and Chee Seong Fong, "An innovative FPGA internal core *clock jitter* prediction methodology," 2010 Asia-Pacific International Symposium on Electromagnetic Compatibility, Beijing, 2010, pp. 346-349.

## DAFTAR RIWAYAT HIDUP PENULIS

### DATA UMUM

Nama Lengkap : A Hadi Syafrudin  
 Tempat & Tgl. Lahir : Bangkalan, 23 Nopember 1980  
 Pangkat / Gol.Ruang : Penata/IIIC  
 Jabatan dalam Pekerjaan : Peneliti Muda  
 NIP : 19801123 200604 1 003



### DATA PENDIDIKAN

SLTA : SMUN 2 Bojonegoro Tahun: 1996  
 STRATA 1 (S.1) : ITS - Surabaya Tahun: 1999 - 2005  
 STRATA 2 (S.2) : IPB - Bogor Tahun: 2008 - 2012

### ALAMAT

Alamat Kantor / Instansi : Pusteksat LAPAN, Jl Cagak Satelit Km.04 Rancabungur Bogor  
 Telp. : 0251-8621667  
 Email : ahaddi@yahoo.com