Seminar 1

IMPLEMENTASI FILTER SPASIAL LINEAR PADA VIDEO STREAM MENGGUNAKAN FPGA HARDWARE ACCELERATOR

Oleh
SULAEMAN
H131 16 002

Pembimbing Utama	: Armin Lawi, M.Eng.
Pembimbing Pertama	: Dr. Diaraya, M.Ak.
Penguji	: 1.Drs. Khaeruddin, MSc
	2.Drs. Aluysius Sutjijana, M.Sc.

PROGRAM STUDI ILMU KOMPUTER
DEPARTEMEN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN
MAKASSAR

2020

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1 Latar Belakang

Citra digital merupakan citra yang dihasilkan dari pengolahan secara digital dengan merepresentasikan citra secara numerik dengan nilai-nilai diskret. Suatu citra digital dapat direpresentasikan dalam bentuk matriks dengan fungsi f(x,y) yang terdiri dari M kolom dan N baris. Perpotongan antara baris dan kolom disebut pixel (Gonzalez and Woods, 2001). Setiap pixel mewakili sebuah warna, pada citra biner sebuah pixel hanya berwarna hitam atau putih saja. Pada citra grayscale warna sebuah pixel mewakili tingkat keabuannya. Sedangkan pada citra warna (RGB) setiap piksel mewakili warna yang merupakan kombinasi dari tiga warna dasar.

Pada umumnya warna dasar dalam citra RGB menggunakan penyimpanan 8 bit untuk menyimpan data warna, yang berarti setiap warna mempunyai gradasi sebanyak 255 warna . Dewasa ini, citra digital dapat menggunakan 16 bit untuk menyimpan data warna dasarnya, hal ini menyebabkan semakin banyak gradasi warnanya sehingga citra yang dihasilkan memiliki tingkat warna yang jauh lebih banyak. Namun tentu saja hal ini mengakibatkan ukuran file citra digital yang dihasilkan juga menjadi semakin besar walaupun dengan resolusi yang sama.

Pengolahan citra digital adalah

Manfaat dari pengolahan citra digital antara lain

Filter Spasial adalah

Video stream dapat dipandang sebagai serangkaian citra digital berturut-turut (Zhao, 2015). Berbeda dengan format video lainya, video stream ini tidak disimpan pada media penyimpanan sebagai file video melainkan langsung disalurkan setiap framenya dari sumber (source) ke penerima, dalam hal ini FPGA. Dengan menganggap Video stream adalah kumpulan citra digital (frame) maka dapat dilakukan metode pengolahan seperti pada citra digital, termasuk filter spasial. Setiap citra yang ditangkap dari source disebut sebagai frame, setiap frame ini dilakukan deteksi tepi kemudian hasilnya ditampilkan secara berkesinambungan sehingga nampak seperti video yang telah difilter.

Frame per second (FPS) atau Frame rate adalah banyaknya frame yang ditampilkan dalam setiap detik pada video. Semakin tinggi FPS sebuah video maka semakin halus pula gerakan yang dapat ditampilkan karena dibentuk dari frame yang lebih banyak, namun dengan jumlah frame yang lebih besar tentu dibutuhkan juga resource yang lebih besar dalam pengolahan video tersebut (Kowalczyk, Przewlocka, and Krvjak, 2018).

Field Programmable Gate Arrays atau FPGA adalah perangkat semikonduktor yang berbasis *matriks configurable logic block* (CLBs) yang terhubung melalui interkoneksi yang dapat diprogram. FPGA dapat diprogram ulang dengan aplikasi atau fungsi yang diinginkan setelah *manufacturing*. Fitur ini yang membedakan FPGA dengan *Application Specific Integrated Circuits* (ASICs), yang dibuat khusus untuk tugas tertentu saja (Xilinx, 2020).

FPGA Xilinx PYNQ-Z2 yang digunakan pada penelitian ini secara official dapat menerima input video stream dari port HDMI Input dengan resolusi maksimal 720p. Video hasil deteksi tepi yang ditampilakan melalui port HDMI Output akan mengalami penurunan fps, hal ini disebabkan oleh penambahan jeda waktu komputasi untuk deteksi tepi antara setiap frame. Setiap frame yang diterima dari *source* akan dilakukan metode deteksi tepi, kemudian hasilnya disalurkan melalui HDMI output untuk kemudian ditampilkan.

1.2 Rumusan Masalah

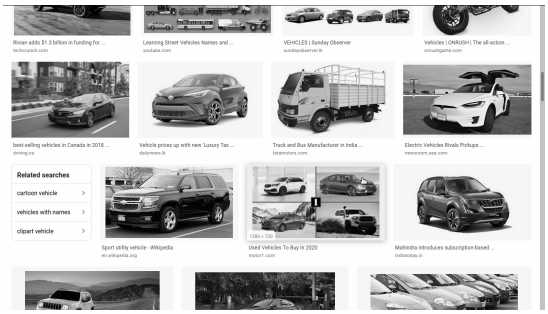
1.3 Batasan Masalah

1.4 Tujuan Penelitian

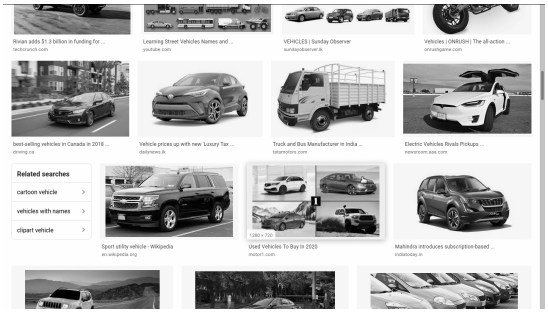
1.5 Manfaat Penelitian

Tabel 1.5.1: Ini Caption tabel

fad	dfaf	fdsfdfads	fdasf	fda
fdas	fdas	ss	ss	ss
ss	dfa	dfsa	fdsa	fdsa
ddd	fdd	dda	da	da



Gambar 1.5.1: Sebuah gambar



Gambar 1.5.2: ini gambar na

BAB II
TINJAUAN PUSTAKA

2.1 Landasan Teori

2.1.1 Citra Digital

Citra digital dapat didefinisikan sebagai fungsi f(x,y) berukuran M baris dan N kolom, dengan x dan y adalah kordinat spasial, dan amplitudo f di titik kordinat (x,y) dinamakan intensitas atau tingkat keabuan dari citra pada citra tersebut (Putra, 2010). Pada umumnya warna dasar dalam citra RGB menggunakan penyimpanan 8 bit untuk menyimpan data warna, yang berarti setiap warna mempunyai gradasi sebanyak 255 warna . Dewasa ini, citra digital dapat menggunakan 16 bit untuk menyimpan data warna dasarnya, hal ini menyebabkan semakin banyak gradasi warnanya sehingga citra yang dihasilkan memiliki tingkat warna yang jauh lebih banyak. Namun tentu saja hal ini mengakibatkan ukuran file citra digital yang dihasilkan juga menjadi semakin besar walaupun dengan resolusi yang sama. Berdasarkan jenis warnanya citra digital dibagi menjadi 3 jenis:

a. Citra Biner (Monokrom)

Banyaknya dua warna, yaitu hitam dan putih. Warna hitam direpresentasikan dengan 1 dan warna putih direpresentasikan dengan 0. Dibutuhkan 1 bit di memori untuk menyimpan warna ini. Contoh citra biner dapat dilihat pada gambar 2.1.1(a).

b. Citra Grayscale (Skala keabuan)

Banyaknya warna tergantung pada jumlah bit yang disediakan di memori untuk menampung kebutuhan warna ini. Citra 2 bit mewakili 4 warna, citra 3 bit mewakili 8 warna, dan seterusnya. Semakin besar jumlah bit warna yang disediakan di memori, semakin halus gradasi warna yang terbentuk. Pada umumnya citra digital grayscale menggunakan 8 bit memori dengan derajat keabuan dari 0 sampai 255. Contoh citra grayscale dapat dilihat pada gambar 2.1.1(b).

c. Citra Warna

Setiap piksel pada citra warna mewakili warna yang merupakan kombinasi dari tiga warna dasar (RG8 = Red Green Blue). Setiap warna dasar menggunakan penyimpanan 8 bit, yang berarti setiap warna mempunyai gradasi sebanyak 255 warna. Berarti setiap piksel mempunyai kombinasi warna sebanyak 255 x 255 x 255 =16 juta warna lebih. Dibutuhkan 3x8 = 24 bit di memori untuk menyimpan sebuah data warna ini. Contoh citra warna dapat dilihat pada gambar 2.1.1(c).

Gambar 2.1.1: (a) Contoh citra biner, (b) contoh citra grayscale, (c) contoh citra warna.

2.1.2 Pengolahan Citra Digital

Pengolahan citra digital merupakan proses mengolah piksel-piksel di dalam citra digital untuk tujuan tertentu. Berdasarkan tingkat pemrosesannya pengolahan citra digital dikelompokkan menjadi tiga kategori, yaitu: *low-level*, *mid-level* dan pemrosesan *high-level*. Pemrosesan *low-level* dilakukan dengan operasi primitif seperti *image preprocessing* untuk mengurangi derau (*noise*), memperbaiki kontras citra dan mempertajam citra (*sharpening*). Karakteristik dari pemrosesan *low-level* yaitu keluaran atau hasil dari pemrosesannya berupa citra digital. Pemrosesan *mid-level* melibatkan tugas-tugas seperti segmentasi (mempartisi gambar menjadi beberapa bagian atau objek), deskripsi objek untuk dilakukan pemrosesan lanjutan, dan klasifikasi objek dalam citra digital. Karakteristik dari pemrosesan *mid-level* yaitu keluaran atau hasilnya berupa atribut atau fitur seperti, kontur, tepi, atau objek yang terdapat dalam citra tersebut. Pemrosesan *high-level* merupakan proses tingkat lanjut dari dua proses

sebelumnya, dilakukan untuk mendapat informasi lebih yang terkandung dalam citra (Gonzalez and Woods, 2001).

2.1.3 Video Streaming

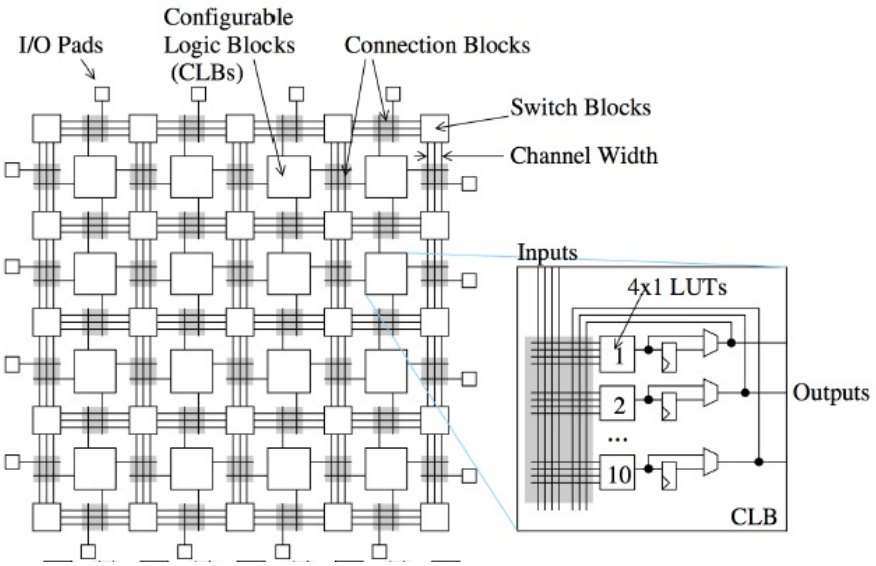
Video stream dapat dipandang sebagai serangkaian citra digital berturut-turut (Zhao, 2015). Berbeda dengan format video lainya, video stream ini tidak disimpan pada media penyimpanan sebagai file video melainkan langsung disalurkan setiap framenya dari sumber (source) ke penerima, dalam hal ini FPGA. Dengan menganggap Video stream adalah kumpulan citra digital (frame) maka dapat dilakukan metode pengolahan seperti pada citra digital, termasuk filter spasial.

2.1.4 FPGA

Field Programmable Gate Arrays atau FPGA adalah perangkat semikonduktor yang berbasis *matriks configurable logic block* (CLBs) yang terhubung melalui interkoneksi yang dapat diprogram.

FPGA dapat diprogram ulang ke aplikasi atau fungsi yang diinginkan setelah *manufacturing*. Fitur ini yang membedakan FPGA dengan *Application Specific Integrated Circuits* (ASICs), yang dibuat khusus untuk tugas tertentu saja (Xilinx, 2020).

Sebuah *microprocessor* menerima instruksi berupa kode 1 atau 0, kode-kode ini selanjutnya diinterpretasikan oleh komputer untuk menjalankan perintah yang diberikan. *Microprocessor* ini membutuhkan intruksi berupa kode secara terus menerus untuk menjalankan fungsinya. Sedangkan pada FPGA hanya dibutuhkan sekali konfigurasi *chip* setiap kali dinyalakan. Membuat atau mengunduh *bitstream* yang menentukan fungsi logika dilakukan oleh *logic elements* (LEs), sebuah sirkuit dapat dibuat dengan mengabungkan beberapa LEs menjadi satu kesatuan. Setelah *bitstream* dipasang, FPGA tidak perlu lagi membaca instruksi berupa 1 dan 0, berbeda dengan *microprocessor* yang selalu membutuhkan instruksi (Cheung, 2019).



Gambar 2.1.2: Struktur FPGA.

2.1.5 Tepi Citra

Suatu titik (x, y) pada citra digital dikatakan sebagai tepi apabila perubahan nilai intensitas derajat keabuan yang mendadak (besar) dalam jarang yang berdekatan. Tepi biasanya terdapat pada batas andara dua daereah yang berbeda pada suatu citra. Tepi pada citra dapat merepresentasikan objek-objek yang terkandung dalam citra tersebut, bentuk dan ukurannya atau terkadang juga informasi tentang teksturnya.

Tepi citra dapat dilihat melalui perubahan intensitas pixel pada suatu area. Berdasarkan perbedaan perubahan intensitas tersebut, tepi dapat dibagi menjadi 4 jenis yaitu tepi steep, ramp, line dan step-line (Putra, 2010).

a. **Steep**, tepi jenis *step* merupakan tepi citra yang berbentuk dari perubahan intensitas nilai pixel secara signifikan dari tinggi ke rendah ataupun sebaliknya.

b. **Ramp**, tepi jenis ini terbentuk dari perubahan intensitas nilai pixel secara perlahan. Perubahan secara perlahan dapat dilihat pada bentuk kurva yang semakin tinggi dengan perubahan kontinu.

c. **Line**, tepi jenis ini ditandai dengan perubahan intensitas nilai pixel secara drastis dari rendah-tinggi-rendah atau sebaliknya.

d. **Step-Line**, tepi *step-line* merupakan gabungan dari tepi jenis step dan line. Tepi

jenis ini ditandai dengan peningkatan intensitas yang tajam dalam interval tertentu dan kemudian ditandai dengan penurunan yang tidak signifikan, sehingga perubahan intensitas nilai pixel selanjutnya berlangsung stabil.

2.1.6 Deteksi Tepi

Deteksi tepi merupakan salah satu metode dalam pemrosesan cira digital untuk deteksi fitur dan ekstraksi dengan mengidentifikasi titik-titik (pixel) dalam citra yang mengalami perubahan tingkat keabuan secara derastis dan mengalami diskontinu. Salah satu tujuan deteksi tepi yaitu untuk mengurangi jumlah data secara signifikan dalam suatu gambar dan mempertahankan sifat strukturalnya untuk pemrosesan citra lebih lanjut. (Herawti and Kardian, 2018).

Berbagai macam metode deteksi dapat digunakan untuk mendeteksi tepi pada citra. Setiap teknik memiliki keunggulan dan karakteristiknya masing-masing. Suatu teknik deteksi tepi mungkin dapat bekerja sangat baik dalam suatu aplikasi tertentu namun sebaliknya belum tentu dapat berjalan secara maksimal pada aplikasi lainnya.

Terdapat berbagai operator deteksi tepi yang telah dikembangkan berdasarkan turunan pertama (first order derrivate), di antaranya operator Robert, operator Sobel, operator Prewitt, operator Krisch, dan operator Canny. Konsep dasar dari deteksi tepi dengan turunan pertama adalah dengan memanfaatkan perbedaan nilai suatu pixel dengan pixel tetangganya.

2.2 Penelitian Terkait

2.2.1 Spatial Filtering Based Boundary Extraction in Underwater Images for Pipeline Detection: FPGA Implementation

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut

metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. (Raj., Abraham, and Supriya, 2016)

2.2.2 FPGA Implementation of Spatial Filtering techniques for 2D Images

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. (Sadangi et al., 2017)

2.2.3 Features of Image Spatial Filters Implementation on FPGA

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. (Ustyukov, Efimov, and Kolchaev, 2019)

2.2.4 An FPGA-Oriented Algorithm for Real-Time Filtering of Poisson Noise in Video Streams, with Application to X-Ray Fluoroscopy

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. (Castellano et al., 2019)

2.2.5 A real-time video denoising algorithm with FPGA implementation for Poisson-Gaussian noise

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. (Tan et al., 2014)

BAB III
METODE PENENILITIAN

3.1 Waktu dan Lokasi Penelitian

3.2 Tahapan Penelitian

3.3 Objek dan Variabel Penelitian

3.4 Instrumen Penelitian

fadfalkfjaldjflk ldf aklfjaldf fasdllfjk (Zhao 2015)

DAFTAR PUSTAKA

[1] G. Castellano et al. “An FPGA-Oriented Algorithm for Real-Time Filtering of Poisson Noise in Video Streams, with Application to X-Ray Fluoroscopy”. In: *Circuits, Systems, and Signal Processing* (Jan. 2019). https://doi.org/10.1007/s00034-018-01020-x.

[2] Peter Cheung. *Introduction to FPGAs*. http://www.ee.ic.ac.uk/pcheung/teaching/ee2_digital/Lecture2-IntroductiontoFPGAs.pdf. Accessed on 2020-04-19. 2019.

[3] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. 2nd. ISBN-13: 978-0201180756. Upper Saddle River, New Jersey 07458: Prentice Hall, 2001.

[4] Desi Herawti and Aqwam Rosadi Kardian. “Analisis Deteksi Tepi Pada Citra Digital Berbasis JPG Dengan Operator Canny Menggunakan Matrix Laboratory”. In: *Jurnal Ilmiah Komputasi* 17.3 (Sept. 2018). p-ISSN 1412-9434/e-ISSN 2549-7227.

[5] Marcin Kowalczyk, Dominika Przewlocka, and Tomasz Krvjak. “Real-Time Implementation of Contextual Image Processing Operations for 4K Video Stream in Zynq UltraScale+ MPSoC”. In: *2018 Conference on Design and Architectures for Signal and Image Processing (DASIP)* (Oct. 2018). DOI: 10.1109/DASIP.2018.8597105.

[6] Darma Putra. *Pengolahan Citra Digital*. ISBN: 978-979-29-1443-6. Jl. Beo 38-40, Yogyakarta 55281: Penerbit Andi, 2010.

[7] S.M. Alex Raj., Rita Maria Abraham, and M.H. Supriya. “Spatial filtering based Boundary Extraction in Underwater Images for Pipeline Detection: FPGA Implementation”. In: *International Journal of Computer Science and Information Security (IJCSIS)* (Sept. 2016). Vol. 14, No. 9.

[8] Sushant Sadangi et al. “FPGA Implementation of Spatial Filtering techniques for 2D Images”. In: *IEEE International Conference On Recent Trends in Electronics Information & Communication Technology (RTEICT)* (May 2017).

[9] Xin Tan et al. “A real-time video denoising algorithm with FPGA implementation for Poisson-Gaussian noise”. In: *J Real-Time Image Proc* (Feb. 2014). https://doi.org/10.1007/s11554-014-0405-2.

[10] Dmitry I. Ustyukov, Alex I. Efimov, and Dmitry A. Kolchaev. “Features of Image Spatial Filters Implementation on FPGA”. In: *MEDITERRANEAN CONFERENCE ON EMBEDDED COMPUTING (MECO)* (June 2019).

[11] Xilinx. *Field Programmable Gate Array (FPGA)*. https://www.xilinx.com/products/silicon-devices/fpga/what-is-an-fpga.html. Accessed on 2020-04-17. 2020.

[12] Jin Zhao. “Video/Image Processing on FPGA”. Master thesis. Worcester Polytechnic Institute, Apr. 2015.