2nd International Conference on Computer Engineering and Science
science without boundary
دومین کنفرانس علوم کامپیوتر
دانشگاه آزاد اسلامی واحد نجف آباد
NOV 15 - 17 ۲۴ الی ۲۶ آبان ۱۴۰۱
Islamic Azad University Najafabad Branch

تبدیل تصاویر RGB به خاکستری با استفاده از واحدهای DSP48E1 تراشهٔ ARTIX-7 با رویکرد افزایش دقت پردازش و کاهش تعداد واحدهای محاسباتی

مهدی عجمین همدانی[۱]، پیام سنائی[۱و۲]

[۱] دانشکده مهندسی برق، واحد نجف آباد، دانشگاه آزاد اسلامی، نجف آباد، ایران
[۲] مرکز تحقیقات پردازش دیجیتال و بینایی ماشین، واحد نجف آباد، دانشگاه آزاد اسلامی، نجف آباد، ایران
* نویسنده مسئول: دکتر پیام سنائی

چکیده

پردازش بی‌درنگ تصاویر دیجیتال یکی از مسایل به روز در علم پردازش تصویر است. حجم بالای تعداد پیکسل‌ها در هر قاب تصویر و نرخ بالای تصویربرداری باعث افزایش پهنای باند اطلاعات تصویری می‌شود. از این رو، کاهش زمان پردازش از جمله چالش‌های مهم محققین است. بسیاری از الگوریتم‌های پردازش تصویر به صورت ذاتی فقط روی تصاویر خاکستری انجام می‌گیرند. همچنین اگر چه برخی از الگوریتم‌های پردازش تصویر توانایی انجام محاسبات بروی تصاویر رنگی را دارا هستند؛ اما برای کاهش حجم حافظهٔ مورد نیاز و کاستن زمان محاسبات، در ابتدا تصاویر رنگی ورودی را به تصاویر خاکستری تبدیل کرده، سپس محاسبات را روی تصاویر خاکستری انجام می‌دهند. از این رو نیاز است تا تصویر رنگی RGB ورودی توسط یک واحد پیش‌پردازش با حداقل تاخیر تبدیل به تصویر خاکستری شود. تراشه‌های مجتمع دیجیتال برنامه‌پذیر (FPGA) یکی از سخت‌افزارهای محبوب برای پیاده‌سازی بی‌درنگ الگوریتم‌های پردازش تصویر به صورت موازی و همروند می‌باشند. عموماً در این نوع از تراشه‌ها محاسبات ریاضی به صورت ممیز ثابت پیاده‌سازی می‌شوند. واحد محاسباتی DSP48E1 یک واحد ریاضی با قابلیت انجام همزمان عمل ضرب و جمع ۴۸ بیتی به صورت ترکیبی است که در تراشه‌های FPGA ساخت شرکت AMD-XILINX موجود است. استفاده از این واحد در مسایل مرتبط با پردازش ریاضی باعث سهولت کار، افزایش دقت محاسبات و کاهش زمان پاسخگویی شده، اما هزینهٔ بالایی داشته و تعداد محدودی از آن در هر تراشهٔ FPGA موجود است. در این مقاله با استفاده از حداقل تعداد واحدهای محاسباتی تصاویر RGB را با حداقل خطا به تصاویر خاکستری تبدیل نموده‌ایم.

کلمات کلیدی

تراشه‌های مجتمع منطقی برنامه‌پذیر، محاسبات ممیز ثابت، واحد ریاضی DSP48E1، رایانش تقریبی، بینایی ماشین، تبدیل تصاویر رنگی به خاکستری

۱- مقدمه

امروزه دانش بازشناسی و شناسایی تصاویر کاربردهای فراوانی در زمینه‌های پزشکی، صنعتی، کشاورزی و نظامی پیدا کرده است. این امر به کمک پیشرفت روزافزون فن‌آوری‌های تصویر برداری دیجیتال و پردازشگرهای محاسباتی دیجیتال سریع میسر شده است. اکثر راه‌کارهای به کار برده شده در برنامه‌های کاربردی شناسایی تصاویر مبتنی بر الگوریتم‌های یادگیری عمیق و روش‌های یادگیری ماشین بوده که خود نیازمند انجام محاسبات سنگین زمانبری هستند. برای حل مسایلی که در آنها با حجم زیادی از اطلاعات برداری مواجه بوده و باید به صورت بی‌درنگ پردازش شوند. عموماً یکی از چهار راهکار سخت‌افزاری زیر استفاده می‌شود.

- پردازشگرهای چند هسته‌ای Multi Core Processor مانند پردازندهٔ مبتنی بر معماری ARM چند هسته‌ای Apple M1 Ultra
- پردازشگرهای ویژهٔ سیگنال‌های دیجیتال Digital Signal Processor مانند پردازشگر تصویر TMS 320C600
- پردازشگرهای گرافیکی Graphic Processor Unit مانند GeForce RTX 3090 Ti
- مدارات مجتمع دیجیتال برنامه‌پذیر FPGA مانند Artix-7

از این میان چهار راهکار فوق مدارات مجتمع دیجیتال برنامه‌پذیر در بین پژوهشگران از مقبولیت بیشتری برخوردار هستند. این امر به دلیل

امکان حل مساله به صورت سخت‌افزاری توسط پیاده‌سازی انواع معماری‌های سخت‌افزاری جهت تسریع محاسبات مانند خط لوله، پردازش موازی و همروند است [۱و۲] . سایر موارد منجر به راهکارهای نرم‌افزاری اجرای برنامه (واکشی، رمزگشایی و اجرای دستور) می‌شوند که نه تنها زمان‌بر بوده، بلکه توان مصرفی بالایی هم دارند. پیاده‌سازی روش‌های سخت‌افزاری غالباً توسط مدارات ASIC/FPGA انجام می‌پذیرد. از این میان تراشه‌های FPGA به دلیل قابلیت پیکربندی مجدد، سادگی پیکربندی و زمان کوتاه پیکربندی متداول‌تر هستند.

در این مقاله هدف تبدیل تصاویر RGB ۲۴بیتی به تصاویر خاکستری ۸ بیتی می‌باشد. یکی از متداول‌ترین روابط برای تبدیل اطلاعات RGB به خاکستری که مبتنی بر دانش رنگ سنجی (Photometric) بوده و توسط موسسه NTSC (National Television Standards Committee) معرفی شده روش میانگین‌گیری وزن‌دهی شده/درخشندگی (Luminosity Method) است [۳]. در این الگوریتم میانگین مولفه‌های رنگی قرمز، سبز و آبی هر پیکسل با ضرایبی غیریکنواخت محاسبه می‌شوند. تجربه نشان داده است که کیفیت بصری تصویر خاکستری تولید شده به روش میانگین‌گیری وزن‌دهی شده نسبت به سایر روش‌ها بهتر بوده و برای ادراک انسان مناسب‌تر است [۳]. این الگوریتم بر اساس این واقعیت است که تراکم سلول‌های مخروطی در شبکیهٔ چشم انسان یکسان نبوده و در چشم انسان حساسیت سلول‌های بینایی نسبت به رنگ سبز بیشتر از رنگ قرمز و رنگ قرمز بیشتر از رنگ آبی است. بنابراین در الگوریتم میانگین‌گیری وزن‌دهی شده به جای برخورد یکسان با مولفه‌های قرمز، سبز و آبی برای هر مولفه، بر اساس میزان حساسیت چشم انسان ضرایب مناسبی را انتخاب می‌نماییم. رابطهٔ (۱) نحوهٔ محاسبهٔ مقدار خاکستری را برحسب مولفه‌های RGB نشان می‌دهد.

$$Gray = R \times ۰,۳۰ + G \times ۰,۵۹ + B \times ۰,۱۱ \qquad (۱)$$

۱. ۱- مروری بر کارهای انجام شده

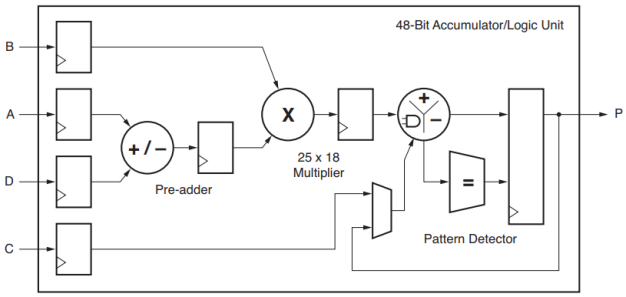
کومار و همکاران روش میانگین‌گیری وزن‌دهی شده را روی تصاویر RGB ۲۴ بیتی مبتنی بر محاسبات شمارشی ۸ بیتی جمع و جابه‌جایی و به صورت سخت‌افزاری روی تراشهٔ FPGA Artix-7 پیاده‌سازی نمودند. اما خطای محاسباتی بسیار بالایی را در قبال کاهش حجم سخت‌افزار ایجاد نموده‌اند[۱]. ژانگ و همکاران یک الگوریتم سریع را برای تبدیل تصاویر RGB به خاکستری با استفاده از روش ضرب ۸ بیتی شمارشی و جمع در سیستم عددی ممیز ثابت ارائه کرده‌اند و آن را توسط تراشهٔ FPGA مدل Spartan-6 پیاده‌سازی نمودند[۲]. یانگ و همکاران الگوریتم فوق را بهینه‌سازی نمودند[۴]. ریوندران یک روش جهت تبدیل تصاویرRGB به خاکستری مبتنی بر دروازه‌های منطقی برگشت‌پذیر ارائه نمودند. ایشان از روش‌های میانگین‌گیری و اشباع‌زدایی برای تبدیل تصاویرRGB به خاکستری استفاده کردند. این طراحی بر روی تراشهٔ FPGA Kintex-7 پیاده‌سازی شده است[۵]. به طور خلاصه، بسیاری از نویسندگان تکنیک‌های مختلفی را برای تبدیل تصاویر RGB به خاکستری پیشنهاد کردند، اما تعداد کمی از آنها به صورت دقیق و با جزییات کامل نحوهٔ پیاده‌سازی سخت‌افزاری الگوریتم‌های پیشنهادی را مورد بحث قرار داده‌اند و عموماً توسط نرم‌افزارهای بالادستی مانند ®Matlab کدها را نوشته و سپس مستقیماً بر روی تراشهٔ FPGA بارگزاری می‌کنند.

۱. ۲- روند پیاده‌سازی سخت‌افزار

ما در این مقاله دو طرح سخت‌افزاری را برای تبدیل تصاویر RGB ۲۴بیتی به خاکستری به روش میانگین‌گیری وزن‌دهی شده ارایه نموده‌ایم؛ که در هر دو از واحدهای DSP48E1 استفاده کرده‌ایم. در سخت‌افزار پیشنهادی اول سیستم عددی به کاربرده شده شمارشی و تمام مثبت بوده و از سه واحد DSP48E1 استفاده کرده‌ایم. طرح سخت‌افزار پیشنهادی دوم بهبود یافتهٔ طرح ابتدایی می‌باشد. در این طرح سخت‌افزاری از ۲ واحد DSP48E1 و یک تمام جمع کنندهٔ ۱۰ بیتی شمارشی و دو تفریقگر شمارشی ۹ بیتی استفاده شده و سیستم عددی به کار رفته شمارشی علامتدار در سیستم مکمل ۲ می‌باشد. در بخش اول به نحوهٔ پیاده‌سازی سخت‌افزارها پرداخته و در ادامه به مقایسهٔ نتایج و بررسی دقت محاسبات و حجم سخت‌افزار خواهیم پرداخت. برای شبیه‌سازی، سنجش و بررسی عملکرد سخت‌افزارهای پیاده‌سازی شده بر روی تراشهٔ FPGA Artix-7، از تصویر رنگی Lenna 512×512 استفاده کرده‌ایم. زبان طراحی سخت‌افزار به کار برده شده VHDL در بستر نرم‌افزار ISE14.7 است.

۲- سخت‌افزارهای پیشنهادی برای پیاده‌سازی الگوریتم میانگین‌گیری وزن‌دهی شده با استفاده از واحدهای DSP48E1

معماری داخلی واحدDSP48E1 تعبیه شده در تراشه‌های مجتمع دیجیتال برنامه‌پذیر خانوادهٔ ARTIX-7 ساخت شرکت AMD-XILINX در شکل ۱ نشان داده شده است [۶]. این واحد در ابتدا برای افزایش قدرت پردازش سیگنال در تراشه‌های FPGA طراحی شده بودند. در نگاه اول این واحد متشکل از یک پیش جمع کنندهٔ ۲۵ بیتی و یک ضرب کنندهٔ مرکزی نسبتاً عریض ۱۸×۲۵ بیتی به همراه یک جمع کنندهٔ انبارهٔ ۴۸ بیتی هستند. این واحد محاسبات را روی اعداد شمارشی علامتدار در سیستم عددی مکمل ۲ انجام می‌دهد. امکان انجام محاسبات به صورت SIMD را دارا بوده و انبارهٔ ۴۸ بیتی می‌تواند عملیات جمع/تفریق/انبارش دو عدد ۲۴ بیتی و یا چهار عدد ۱۲ بیتی را همزمان انجام دهد.



شکل ۱: معماری داخلی واحد محاسباتی DSP48E1 [۶]

۲. ۱- سخت‌افزار پیشنهادی اول

برای پیاده‌سازی رابطهٔ (۱) در این طرح نیاز به ۳ واحد محاسباتی DSP48E1 خواهیم داشت. هر واحد DSP48E1 یکی از مولفه‌های رنگی پیکسل را در ضریب متناظر ضرب می‌کند. از این رو در ابتدا باید محاسبات اعشاری را به محاسبات شمارشی تبدیل نماییم. از آنجایی که در واحد DSP48E1 یکی از عملوندهای ورودی ضرب کننده ۱۸ بیتی و عملوند دیگر ۲۵ بیتی است. اطلاعات نظیر مولفه‌های رنگی RGB هشت بیتی را به صورت ۱۸ بیتی و علامتدار تبدیل کرده و ضرایب را به صورت اعداد ۲۵ بیتی شمارشی علامتدار در آورده تا حداکثر دقت ممکنه در محاسبات کسب گردد. بدلیل این که ضرایب، اعداد مثبتی بوده بیت با ارزش بیشتر عملوند ۲۵ بیتی صفر است. همچنین برای رسیدن به حداکثر فضای پوششی ۲۵ بیتی کلیهٔ ضرایب اعشاری را در عدد $۲^{۲۴}$ ضرب کرده و با استفاده از روش گرد کردن اعداد اعشاری را به اعداد شمارشی تبدیل می‌نماییم. رابطهٔ (۱) را می‌توان به صورت مجموعه روابط زیر بیان کرد.

$$Gray = \frac{(R \times ۰,۳۰ + G \times ۰,۵۹ + B \times ۰,۱۱) \times ۲^{۲۴}}{۲^{۲۴}}$$

$$Gray = \frac{(R \times ۰,۳۰ \times ۲^{۲۴} + G \times ۰,۵۹ \times ۲^{۲۴} + B \times ۰,۱۱ \times ۲^{۲۴})}{۲^{۲۴}}$$

$$۰,۳۰ \times ۲^{۲۴} = ۵۰۳۳۱۶۴,۸۰ \approx ۵۰۳۳۱۶۵ = ۰\_۰۱۰۰\_۱۱۰۰\_۱۱۰۰\_۱۱۰۰\_۱۱۰۱$$

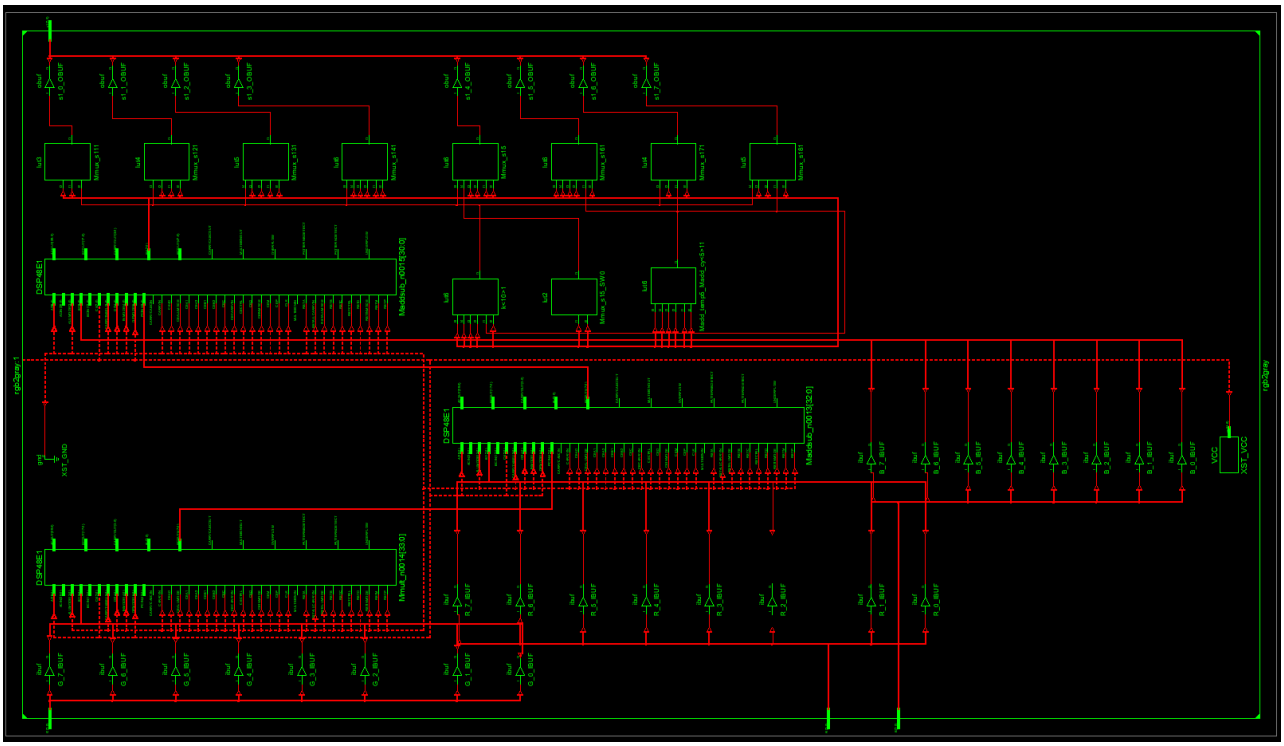
$$۰,۵۹ \times ۲^{۲۴} = ۹۸۹۸۵۵۷,۴۴ \approx ۹۸۹۸۵۵۷ = ۰\_۱۰۰۱\_۰۱۱۱\_۰۰۰۰\_۱۰۱۰\_۰۰۱۱\_۱۱۰۱$$

$$۰,۱۱ \times ۲^{۲۴} = ۱۸۴۵۴۹۳,۷۶ \approx ۱۸۴۵۴۹۴ = ۰۰۰۰۱۰۱۱۰۰۰۰۱۰۱۰۰۰۱۱۱۱۰۱۱۰$$

$$\Rightarrow Gray = \frac{(R \times ۵۰۳۳۱۶۵ + G \times ۹۸۹۸۵۵۷ + B \times ۱۸۴۵۴۹۴)}{۲^{۲۴}} \quad (۲)$$

اکنون محاسبات صورت رابطهٔ (۲) را به صورت شمارشی انجام داده، سپس برای تقسیم حاصل محاسبات بر $۲^{۲۴}$، از جابه‌جایی به سمت راست استفاده می‌کنیم و ۲۴بیت با ارزش کمتر صورت کسر را حذف می‌نماییم. برنامهٔ VHDL این طرح سخت‌افزاری به شرح زیر است. در نهاد این کد، درگاه‌های ورودی R,G,B اطلاعات ۸ بیتی متناظر با مولفه‌های رنگی هر پیکسل بوده و درگاه خروجی s1 مقدار خاکستری نظیر مولفه‌های رنگی بر اساس رابطهٔ درخشندگی است. سه سیگنال temp1,temp2,temp3 ۴۳ بیتی بوده و حاصل ضرب مقادیر مولفه‌های RGB در ضرایب نظیر را در آنها ذخیره می‌کنیم. شایان ذکر است که با توجه به ۸ بیتی بودن مولفه‌های رنگی حداقل ۱۰ بیت با ارزش بیشتر این سیگنال‌ها حتماً صفر می‌باشند. این امر باعث شده تا از عدم وقوع سرریز به هنگام جمع سه سیگنال temp1,temp2,temp3 با یکدیگر اطمینان حاصل نماییم. حال این سه سیگنال را با یکدیگر جمع کرده و در سیگنال temp4 ذخیره می‌کنیم. در واقع اکنون صورت رابطهٔ (۲) در سیگنال temp4 قرار گرفته است. برای تقسیم بر $۲^{۲۴}$ از ۴ بیت با ارزش کمتر سیگنال صرفه نظر می‌کنیم. اما اگر اولین بیت اعشاری با وزن $۲^{-۱}$ یک باشد باید حاصل به سمت بالا گرد شود و یک واحد شمارشی به عدد افزوده شود. در خط ۲۲ از برنامه این امر محقق شده است. حال اگر یکی از بیت‌های temp5(19 downto 9) یک باشد حاصل محاسبات بیشتر از عدد ۲۵۵ شده و باید عمل اشباع شدن را انجام دهیم. در غیر این صورت ۸ بیت کم ارزش temp5(8 downto 1) معادل خروجی محاسبه خواهد شد. در شکل ۲ مدار معادل طراحی نمایش داده شده است. برای تست عملکرد سخت‌افزار پیشنهادی اول در ابتدا فایل تصویری LENNA 512×۵۱۲ رنگی را به سخت‌افزار اعمال کرده و نتایج را با خروجی برنامهٔ نوشته شده به زبان MATLAB® و محاسبات ممیز شناور مقایسه کرده و معیارهای حداکثر قدرمطلق خطا و MSE دو تصویر را بدست آوردیم. همچنین در برنامهٔ آزمونهٔ VHDL توسط سه حلقهٔ تو در تو تمامی حالت‌های سه تایی ممکنه را برای مولفه‌های رنگیRGB را اعمال کرده و خروجی خاکستری ایجاد شده را با محاسبات ممیز شناور مقایسه می‌نماییم.

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  use IEEE.NUMERIC_STD.ALL;
4
5  entity rgb2gray is
6      Port ( R           : in   STD_LOGIC_VECTOR (7 downto 0);
7             G           : in   STD_LOGIC_VECTOR (7 downto 0);
8             B           : in   STD_LOGIC_VECTOR (7 downto 0);
9             s1          : out  STD_LOGIC_VECTOR (7 downto 0));
10 end rgb2gray;
11
12 architecture Behavioral of rgb2gray is
13   signal temp1,temp2,temp3 : signed(42 downto 0);
14   signal temp4             : signed(42 downto 0);
15   signal temp5             : signed(19 downto 0);
16   signal k                 : STD_LOGIC_VECTOR(19 downto 9);
17 begin
18    temp1 <= to_signed(5033165,25) * signed(("0000000000"&R));
19    temp2 <= to_signed(9898557,25) * signed(("0000000000"&G));
20    temp3 <= to_signed(1845494,25) * signed(("0000000000"&B));
21    temp4 <= temp1 + temp2 + temp3;
22    temp5 <= temp4(42 downto 23) + "01";-- rounding no need to saturate
23    --
24    k(9) <= temp5(9);
25    l1: for i in 10 to 19 generate
26         k(i) <= k(i-1) or temp5(i);
27    end generate;
28    --
29    s1    <= "11111111" when k(10) = '1'
30                        else std_logic_vector(temp5(8 downto 1));
31 end Behavioral;
```



شکل ۲: معماری داخلی سخت‌افزار پیشنهادی اول

۲. ۲- سخت‌افزار پیشنهادی دوم

در روش دوم برای کاهش تعداد واحدهای محاسباتی DSP48E1 به کار رفته از یک لِم ریاضی استفاده می‌کنیم. از آنجایی که مجموع ضرایب مولفه‌های رنگی در رابطهٔ (۱) برابر با یک می‌باشد. می‌توان یک ضریب را بر حسب دو ضریب دیگر محاسبه نمود. روند پیشنهادی در مجموعه روابط زیر آورده شده است.

$$۱ \quad = ۰,۳۰ + ۰,۵۹ + ۰,۱۱ \tag{۳}$$

$$۰,۱۱ \quad = ۱ \quad - ۰,۳۰ - ۰,۵۹ \tag{۴}$$

$$\Rightarrow Gray \; = R \times ۰,۳۰ + G \; \times ۰,۵۹ + B \times (۱ - ۰,۳۰ - ۰,۵۹) \tag{۵}$$

$$\Rightarrow Gray \; = (R - B) \times ۰,۳۰ + \; (G - B) \times ۰,۵۹ + B \tag{۶}$$

همانگونه که در رابطهٔ (۱) مشاهده می‌شود برای انجام محاسبات نیاز به ۳ عمل ضرب و ۲ عمل جمع خواهیم داشت در صورتی که رابطهٔ (۶) شامل دو عمل ضرب، دو تفریق و دو جمع است. اگر چه تعداد عملیات ریاضی در مقایسه با رابطهٔ (۱) افزایش یافته اما نوع عملیات و ابعاد عملوندها متفاوت است. تفریقگرهای به کار برده شده ۹ بیتی و شمارشی بوده و یکی از جمع کننده‌ها ۴۳ بیتی و جمع کنندهٔ دیگر ۸ بیتی است. حال از رابطهٔ (۶) می‌توانیم روابط زیر را بدست آوریم.

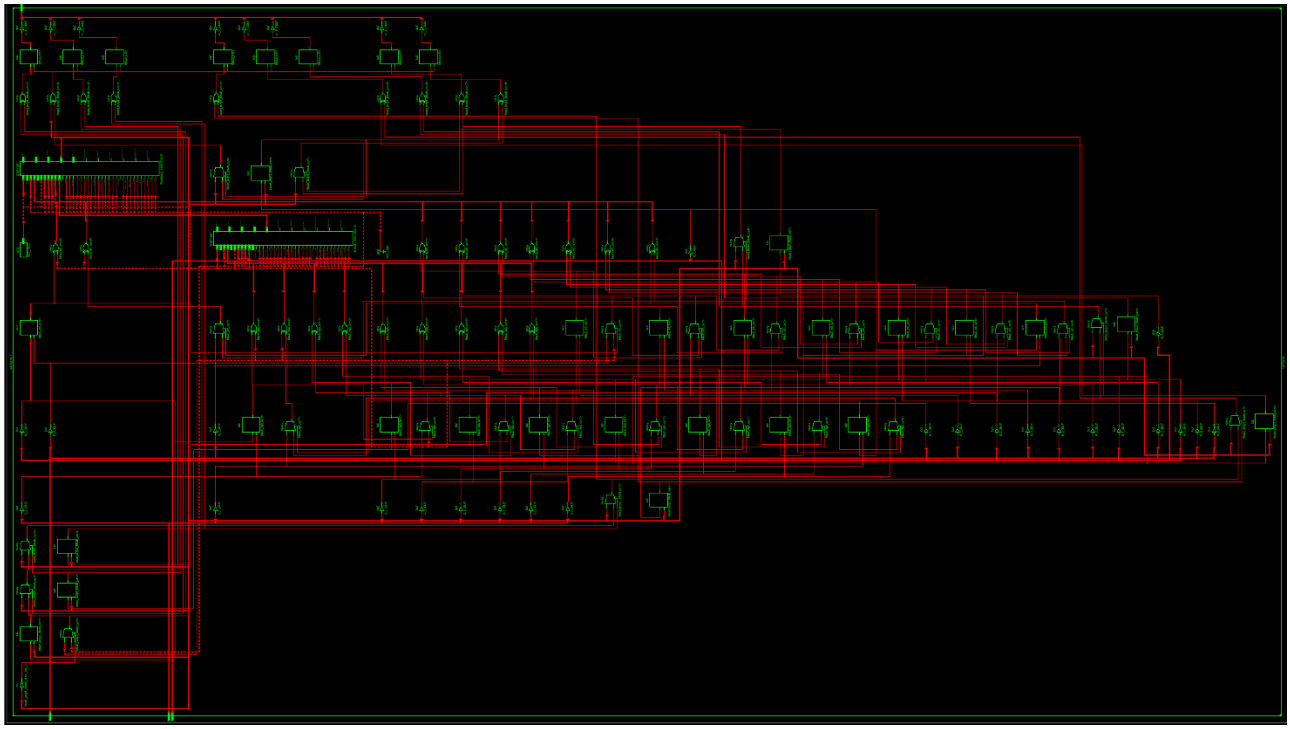
$$\Rightarrow Gray \; = \frac{((R - B) \times ۰,۳۰ + \; (G - B) \times ۰,۵۹) \; \times \; ۲^{۲۴}}{۲^{۲۴}} + B \tag{۷}$$

$$\Rightarrow Gray \; = \frac{((R - B) \times ۰,۳۰ \times \; ۲^{۲۴} + \; (G - B) \times ۰,۵۹ \times \; ۲^{۲۴})}{۲^{۲۴}} + B \tag{۸}$$

$$\Rightarrow Gray \; = \frac{((R - B) \times ۵۰۳۳۱۶۵ + (G - B) \times ۹۸۹۸۵۵۷)}{۲^{۲۴}} + B \tag{۹}$$

در ادامه برنامهٔ VHDL مربوط به پیاده‌سازی طرح سخت‌افزاری دوم به همراه معماری داخلی این طرح در شکل ۳ نمایش داده شده است. در جدول ۱ منابع سخت‌افزاری به کار برده شده در هر روش و دقت محاسبات برای بررسی کمی نمایش داده شده‌اند. همچنین برای بررسی کیفی تصویر Lenna 512×۵۱۲ رنگی توسط هر دو سخت‌افزار پیشنهادی به تصاویر خاکستری تبدیل شده و به همراه تصویر خاکستری ایجاد شده توسط محاسبات ممیز شناور در شکل ۴ نمایش داده شده‌اند.

```
 1  library IEEE;
 2  use IEEE.STD_LOGIC_1164.ALL;
 3  use IEEE.NUMERIC_STD.ALL;
 4
 5  entity rgb2gray is
 6      Port ( R           : in   STD_LOGIC_VECTOR (7 downto 0);
 7             G           : in   STD_LOGIC_VECTOR (7 downto 0);
 8             B           : in   STD_LOGIC_VECTOR (7 downto 0);
 9             s1          : out  STD_LOGIC_VECTOR (7 downto 0));
10  end rgb2gray;
11
12  architecture Behavioral of rgb2gray is
13    signal srb,sgb,sb        : signed( 8 downto 0);
14    signal temp1,temp2,temp3 : signed(42 downto 0);
15    signal temp4             : signed(10 downto 0);
16  begin
17     sb    <= signed('0' & B)     ;
18     srb   <= signed('0' & R) - sb;
19     sgb   <= signed('0' & G) - sb;
20     temp1 <= to_signed(5033165,25) * resize(srb,18);
21     temp2 <= to_signed(9898557,25) * resize(sgb,18);
22     temp3 <= temp1 + temp2;
23     temp4 <= temp3(33 downto 23) + ('0' & sb & '1');-- rounding no need to saturate
24     s1    <= "11111111" when (temp4(9) = '1')
25                          else std_logic_vector(temp4(8 downto 1));
26  end Behavioral;
```



شکل ۳: معماری داخلی سخت‌افزار پیشنهادی دوم

شکل ۴: آ)تصویر رنگی Lenna 512×۵۱۲، ب) تصویر خاکستری شده توسط محاسبات ممیز شناور در بستر زبان Matlab، پ) تصویر خاکستری شده توسط سخت‌افزار پیشنهادی اول، ت) تصویر خاکستری شده توسط سخت‌افزار پیشنهادی دوم، ث) تصویر خاکستری شده توسط روش مرجع [۱]، ج) تصویر خاکستری شده توسط روش مرجع [۲]، چ) تصویر خاکستری شده توسط روش مرجع [۴]، ح) تصویر خاکستری شده توسط روش مرجع [۵]

جدول ۱: اندازه و نوع قلم‌ها

روش پیشنهادی	منابع سخت‌افزاری به کار برده شده		Lenna 512×۵۱۲		۰ : ۲۵۵ R,G,B	
	DSP48E1	LUT	Max abs-err	MSE	Max abs-err	MSE
روش پیشنهادی اول	۳	۱۱	۱	۰.۰۰۲۴	۱	۰.۰۰۴۵
روش پیشنهادی دوم	۲	۳۳	۱	۰.۰۰۲۴	۱	۰.۰۰۴۵
روش مرجع [۱]	-	۳۴	۸	۱۷.۴۲۲	۹	۱۹.۵۶۰
روش مرجع [۲]	۳	۴۵	۳	۰.۰۲۹۶	۱	۰.۰۷۵۷
روش مرجع [۴]	۳	۹۳	۱	۰.۰۴۹۵	۱	۰.۰۶۲۵
روش مرجع [۵]	-	۱۰۴	۱	۵.۵۰۴۵	۴	۷.۳۶۵

۳- نتیجه‌گیری و جمع‌بندی

استفاده از واحد محاسباتی قدرتمند DSP48E1 در تراشه های مجتمع منطقی برنامه‌پذیر باعث افزایش سرعت محاسبات و کاهش واحدهای LUT می‌گردد. این واحدها خود گران قیمت بوده و استفادهٔ بهینه از این واحدها خود امری بسیار مهم می‌باشد. تا مصالحه‌ای بین دقت و هزینه در طراحی های سخت‌افزاری ایجاد شود. در این مقاله دو راهکار سخت‌افزاری مبتنی بر واحدهای محاسباتی DSP48E1 برای تبدیل تصاویر رنگی ۲۴ بیتی به خاکستری مبتنی بر متد میانگین‌گیری وزن‌دهی‌شده ارائه کردیم. دقت محاسبات در هر دو روش بسیار بالا و یکسان بوده اما تعداد واحد سخت‌افزاری DSP48E1 به کار برده شده در روش دوم یک واحد کمتر از از روش اول می‌باشد. برای ادامهٔ پژوهش پیشنهاد می‌شود تا بررسی شود که ایجاد کدام ضریب بر اساس ضرایب دیگر باعث افزایش دقت محاسبات خواهد شد. در طرح سخت‌افزاری دوم ضریب ۰,۱۱ بر اساس دو ضریب ۰,۳۰ و ۰,۵۹ ساخته شد. این پژوهش مستخرج از طرح تحقیقاتی به شمارهٔ ۳۹۷۱/ص/۱۴۰۰فی مابین دانشگاه آزاد اسلامی واحد نجف‌آباد و گروه فن‌آوران سپاهان رمزینه می‌باشد [۷].

۴- مراجع

[1] Kumar, K., Mishra, R.K., and Nandan, D., “Efficient Hardware of RGB to Gray Conversion Realized on FPGA and ASIC”. Procedia Computer Science, No 171, pp. 2008-2015, 2020.

[2] Zhang, Y., Yang, X., Wu, L., Lu, J., Sha, K., Gajjar, A., and He, H., “Exploring slice-energy saving on an video processing FPGA platform with approximate computing”. In Proceedings of the 2018 2nd International Conference on Algorithms, Computing and Systems, China, (pp. 138-143), July 2018.

[3] Gonzalez RC, Woods RE; *Digital Image Processing, Global Edition.* Fourth Edition, Pearson, 2018.

[4] Yang, X., and Sha, S., “Exploiting Energy–Quality (E–Q) Tradeoffs: A Case Study on Color-to-Grayscale Converters with Approximate Design on FPGA”. Journal of Circuits, Systems and Computers, No 30(04), pp.2150062. 2021

[5] Raveendran, S., Edavoor, P.J., Kumar, N.Y., and Vasantha, M.H., “Design and Implementation of Reversible Logic based RGB to Gray Scale Color Space Converter”, *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, pp. ۱۸۱۳-۱۸۱۷. ۲۰۱۹.

[6] https://docs.xilinx.com/v/u/en-US/ug479_7Series_DSP48E1

[۷] سنائی، پیام؛ **طراحی و ساخت بورد FPGA جهت تشخیص لبه در تصاویر مبتنی بر الگوریتم SOBEL و بهینه سازی آن با سه رویکرد کاهش حجم سخت افزار به کار برده شده، افزایش سرعت محاسبات و افزایش دقت** ، طرح تحقیقاتی شمارهٔ ۳۹۷۱/ص/۱۴۰۰، دانشگاه‌آزاد اسلامی واحد نجف‌آباد به سفارش گروه فن‌آوران سپاهان رمزینه، اصفهان، ۱۴۰۰.