

طراحی کامپیوتری سیستم‌های دیجیتال

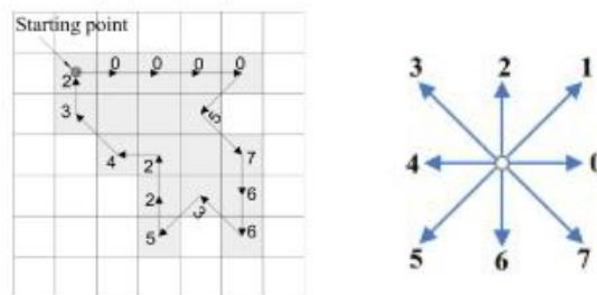
نیمسال اول ۱۴۰۲-۱۴۰۳

پروژه پایانی

الگوریتم‌های کد کردن تصویر با استفاده از بخش بندی تصویر تلاش می کنند تا خصوصیات بخش های مهم و شناخته شده تصویر را به گونه‌ای موثر و به نحوی که بازسازی مجدد تصویر با دقت کافی امکان پذیر باشد، توصیف کنند. الگوریتم‌های متعددی برای کد کردن تصویر وجود دارد. الگوریتم chain code یکی از الگوریتم‌های کد کردن تصویر می باشد.

در این الگوریتم که برای کد کردن لبه های یک شی استفاده می‌شود، هریک از پیکسل های لبه شی با یکی از ۸ بردار جهت نشان داده شده در شکل ۱ بیان می شوند. به این ترتیب با تخصیص کد به هریک از این بردارهای جهت می توان لبه یک شی را کدگذاری کرد.

عملکرد الگوریتم به این صورت است که پس از یافتن اولین پیکسل لبه شی موجود در تصویر، جهت حرکت به سمت پیکسل بعدی روی لبه شی در خلاف جهت عقربه های ساعت با توجه به یکی از ۸ جهت موجود تعیین گردیده و ذخیره می گردد. پس از آن با اجرای الگوریتم روی پیکسل بعدی، جهت حرکت به سمت پیکسل بعد از آن تعیین می گردد. این کار آنقدر ادامه می یابد تا مجدداً به پیکسل ابتدایی باز گردیم.



شکل ۱، جهت‌ها در الگوریتم Chain Code و یک مثال از انجام کدینگ

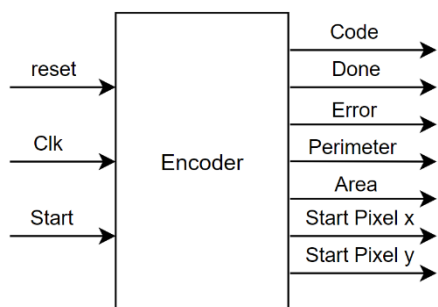
نکات مهم:

۱. همواره باید با توجه به لبه های تصویر ، اولویت لبه ها برای کد کردن تصویر با توجه به حرکت در خلاف جهت حرکت عقربه های ساعت و با برتر قرار دادن لبه های راست و بالا به ترتیب نسبت به سایر لبه ها حرکت و کدگذاری انجام شود. جهت ۰ دارای بیشترین اولویت و جهت هفت دارای کمترین اولویت است.
۲. در تمامی اشکال فرض بر این است که جسم کاملاً توپر بوده و هیچ فضای خالی در داخل جسم بوسیله پیکسل های روشن احاطه نشده است.

۳. در تمامی اشکال تنها یک شی در داخل تصویر وجود دارد.

۴. الگوریتم بر روی تصویرهای باینری پیاده‌سازی می‌شود.

در این پروژه هدف پیاده‌سازی یک فرستنده و گیرنده با ارتباط Uart است. در سمت فرستنده یک Chain Code Encoder همانند شکل زیر وجود دارد. Encoder پس از فعال شدن start و لبه‌ی بالارونده‌ی کلاک، مقادیر باینری تصویر را از حافظه خوانده و شروع به کد کردن تصویر مینماید.



جزئیات پورت های ورودی و خروجی به شرح زیر است:

الف) ورودی ها:

Reset: ریست کردن ماژول

CLK: کلاک اصلی سخت افزار

Start: شروع کار ماژول بر روی تصویری که از قبل در blk memory قرار گرفته است.

ب) خروجی ها:

Code: کد خروجی تولید شده

Done: فرمان اتمام عملیات کد گذاری که پس از اتمام کد گذاری ۱ خواهد شد.

Error: اعلام وجود یا عدم وجود مشکل و خطایی در هر یک از عملیات های پردازش تصویر که در صورت وقوع خطا ۱ خواهد شد.

Perimeter: محیط تعیین شده برای تصویر داده شده بر اساس اندازه لبه‌ی هر پیکسل

Area: مساحت تصویر داده شده (تعداد پیکسل‌های روشن تصویر)

Start pixel: مختصات پیکسلی که به عنوان اولین لبه‌ی تصویر توسط ماژول تشخیص داده شده است.

پس از پایان Encoding کد خروجی در قالب ارتباط سریال از نوع UART به سمت گیرنده ارسال گردد. مشخصات مورد نیاز برای ارتباط سریال از نوع UART به شرح زیر خواهد بود:

Clock frequency: 50 Mhz

در سمت گیرنده، یک Chain Code Decoder همانند شکل زیر وجود دارد. Decoder پس از فعال شدن start و لبه‌ی بالارونده‌ی کلاک، شروع به رمزگشایی و بازسازی تصویر دریافت شده بر اساس کد می‌نماید.

جزئیات پورت های ورودی و خروجی به شرح زیر است:

الف) ورودی ها:

Reset: ریست کردن ماژول

CLK: کلاک اصلی سخت افزار

Start: شروع کار ماژول بر روی تصویری که از قبل در blk memory قرار گرفته است.

Code: کد ورودی دریافت شده.

Perimeter: محیط تعیین شده برای تصویر داده شده بر اساس اندازه لبه‌ی هر پیکسل

Area: مساحت تصویر داده شده (تعداد پیکسل‌های روشن تصویر)

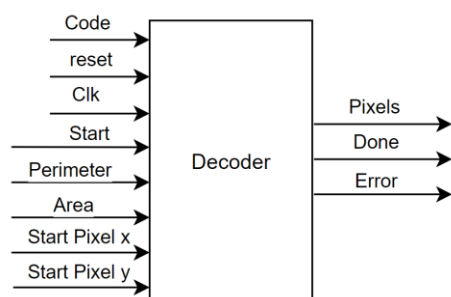
Start Pixel: مختصات پیکسلی که به عنوان اولین لبه‌ی تصویر توسط ماژول تشخیص داده شده است.

(ب) خروجی‌ها:

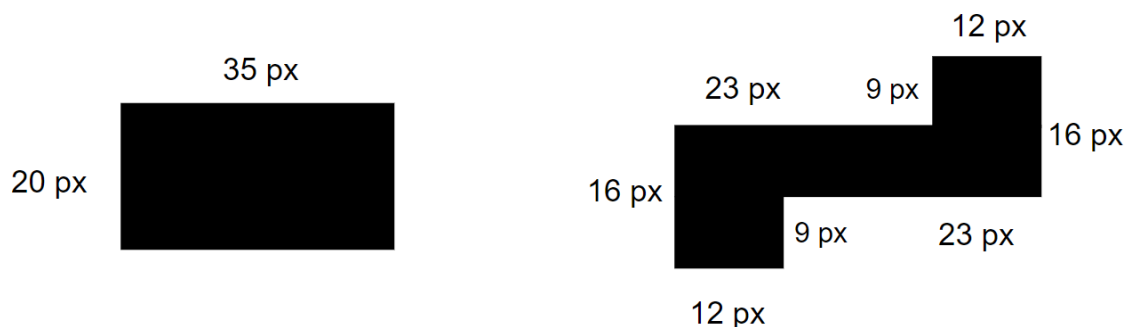
Done: فرمان اتمام عملیات کد گذاری که پس از اتمام کد گذاری ۱ خواهد شد.

Error: اعلام وجود یا عدم وجود مشکل و خطایی در هر یک از عملیات‌های پردازش تصویر که در صورت وقوع خطا ۱ خواهد شد.

Pixels: رسم تصویر اولیه از طریق ۱ کردن پیکسل‌هایی که روشن هستند (دو بعدی)



تصاویر نمونه برای تشخیص صحت سخت افزار توصیف شده به شرح زیر می‌باشد.



شکل ۲، نمونه‌ی تصاویر برای صحت سنجی فرستنده و گیرنده

عکس‌ها باید توسط زبان برنامه‌نویسی دیگری مانند متلب یا پایتون به باینری تبدیل شوند. مقادیر پیکسلی تصویر میبایست ابتدا در یک فایل قرار گرفته و به هنگام توصیف سخت افزار همگی در یک ipcore از نوع ram قرار بگیرند. لذا رعایت ترتیب ذخیره

سازی پیکسل ها به هنگام خواندن از ipcore مهم خواهد بود. داده های ورودی ماژول Encoder نیز در هر لحظه با توجه به نیاز از ipcore خوانده می شوند.

اندازه ی تصویر ورودی ۶۴*۶۴ پیکسل بوده و انتخاب عرض بیت های مورد نیاز ورودی و خروجی ماژول و ipcore دلخواه می باشد.

تحويل پروژه

- ۱- با توجه به ساختار Top Down Design، تمامی زیر ماژول های مناسب برای فرستنده، گیرنده و Uart را پیاده سازی کنید.
- ۲- ارسال گزارش به همراه فایل های پروژه الزامی است و در صورت نبود هر کدام پروژه بررسی نمی گردد. گزارش ارسالی شامل توضیحاتی درباره ی هر کدام از زیرماژول ها، نحوه ی کارکرد آن ها و تصویر simulator از خروجی می باشد. فایل های پروژه شامل کد زیر ماژول های تعریف شده و testbench می باشد.
- ۳- گزارش پروژه را در Quera درس آپلود کنید.
- ۴- پروژه به صورت گروه های دو نفره میتواند انجام داده شود (در این صورت در فایل گزارش نام افراد گروه ذکر شده و قسمت های انجام شده توسط هر فرد نیز مشخص شود)
- ۵- تاریخ تحويل پروژه ۱۲ و ۱۳ بهمن ماه ۱۴۰۲ خواهد بود.
- ۶- تحويل پروژه به صورت آنلاین بوده و لینک ارائه در روزهای آینده اطلاع رسانی خواهد شد. تحويل پروژه به صورت فردی است و هر فرد باید بتواند تمامی ماژول های سمت فرستنده و گیرنده و Uart را شرح دهد.

توضیحات و پیشنهادهایی برای پیاده سازی پروژه

این سخت افزار به صورت State Machine عمل می کند، لذا انجام هر کاری در آن وابسته به لبه های بالا رونده ورودی CLK استاین ماشین حالت سه وضعیت دارد که در state اول، مقدار دهی اولیه انجام می گیرد. در state دوم یافتن اولین پیکسل روشن در لبه یا اولین لبه شی انجام می شود و در سومین state کدهای مربوط به پیکسل های لبه ی شی تا بازگشت به اولین پیکسل لبه تولید می شود.

در این سخت افزار از یک بیت خروجی به نام Error _ Flag برای نشان دادن هرگونه خطا در دیتای ورودی استفاده می شود. در صورتی که مقدار آن ۱ باشد، نشان دهنده وقوع خطا در تصویر می باشد. اگر پس از وارد کردن دیتای کلیه پیکسل ها با توجه به ابعاد تصویر، هیچ پیکسل با مقدار ۱ پیدا نشود، Error _ Flag به نشانه خطا در تصویر ۱ می گردد. همچنین اگر در هر یک از مراحل حرکت بر روی لبه تصویر مشکلی در حرکت بوجود بیاید مجددا این بیت به نشانه خطا در کد کردن لبه تصویر ۱ خواهد شد.

این سخت افزار کل تصویر را در خود ذخیره خواهد کرد، باید در هنگام وارد کردن اطلاعات مربوط به همسایه های پیکسل جاری به این نکته توجه کرد که اگر پیکسل جاری جز کناره تصویر می باشد و پیکسل های همسایه که در خارج از تصویر هستند را ۰ در نظر می گیریم.