



دانشکده برق و کامپیوتر

دستور کار پروژه نهایی سوم زبان توصیف سخت افزار

نیمسال دوم سال تحصیلی 1398-99

آخرین مهلت زمان تحویل: متعاقبا اعلام می‌گردد

نکاتی در مورد انجام و ارسال پروژه:

1. طراحی باید کاملاً ماژولار (سلسله مراتبی) صورت گیرد.
2. در کنار هر ماژول، یک ماژول testbench نیز لازم است.
3. توجه فرمایید که نیازی به ارسال همه‌ی فایل‌های پروژه‌ی نرم‌افزار ISE نیست! فقط فایل‌های v. را در کنار گزارش‌تان ارسال فرمایید.
4. برای ارسال نتایج شبیه‌سازی، از ارسال فایل خروجی نرم‌افزارهای شبیه‌ساز خودداری فرمایید. نتایج راتنها به شکل screenshot با کیفیت و جزئیات قابل قبول ارسال فرمایید.
5. در صورت وجود سوال و یا مشکل در انجام پروژه، از طریق ایمیل و یا اکانت تلگرام تماس برقرار کنید.
6. در نهایت، یک فایل زیپ به شکل HW_final_studentID.zip متناظر با آیدی سرگروه تیم، حاوی گزارش و فایل‌های v. بایستی تحویل دهید.

در این پروژه می‌خواهیم یک فایل تصویر را از یک حافظه ROM فراخوانی و با انجام یک عملیات پردازشی بر روی آن، لبه‌های موجود در تصویر را آشکارسازی نماییم. برای پیاده‌سازی این آشکارساز لبه، از الگوریتم ساده **sobel** استفاده می‌کنیم. در این الگوریتم ابتدا ماتریس‌های عملگر در راستای محور افقی و عمودی را به فرم زیر تعریف می‌کنیم:

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, \quad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

که در آن ماتریس G_x (G_y)، تغییرات موجود در تصویر را در راستای افقی (عمودی) استخراج می‌کند. برای به دست آوردن این تغییرات، کافی است این ماتریس‌ها را بر روی تصویر اصلی در راستای افقی و عمودی حرکت داده و با جمع کردن حاصلضرب مقادیر هم‌پوشانی شده از تصویر اصلی با مقادیر ماتریس‌های فوق، به هر پیکسل یک **ضریب وزنی** اختصاص دهیم (شکل 1). بدین ترتیب در نهایت دو ماتریس A_x و A_y خواهیم داشت. که با تشکیل یک ماتریس $A = |A_x| + |A_y|$ ، می‌توان لبه‌های موجود در تصویر را با انتخاب یک سطح آستانه‌گذاری مناسب به صورت تقریبی پیدا نمود. در شکل 2 روند کلی الگوریتم برای محور افقی نمایش داده شده است. **توجه شود** که هنگامی که ماتریس G_x به انتهای هر ستون از تصویر می‌رسد آن را به اول سطر بعد انتقال داده و مجدداً حرکت می‌دهیم. این عمل ممکن است باعث شود که در سطرها و ستون‌های ابتدایی نتوانیم لبه‌ها را آشکارسازی نماییم که البته چندان اهمیتی نخواهد داشت.

1	0	-1
2	0	-2
1	0	-1

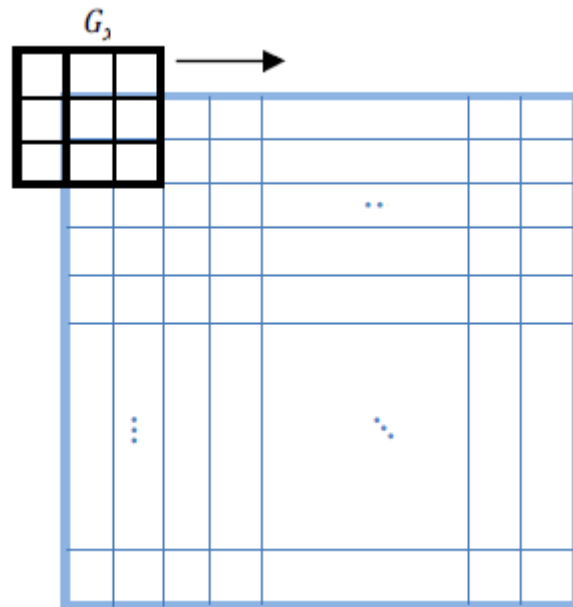
a_{11}	a_{12}	a_{13}
a_{21}	a_{22}	a_{23}
a_{31}	a_{32}	a_{33}

$$(pixel\ weight)_{center} =$$

$$a_{11} - a_{13} + 2a_{21}$$

$$- 2a_{23} + a_{31} - a_{33}$$

شکل 1- نحوه تعیین وزن

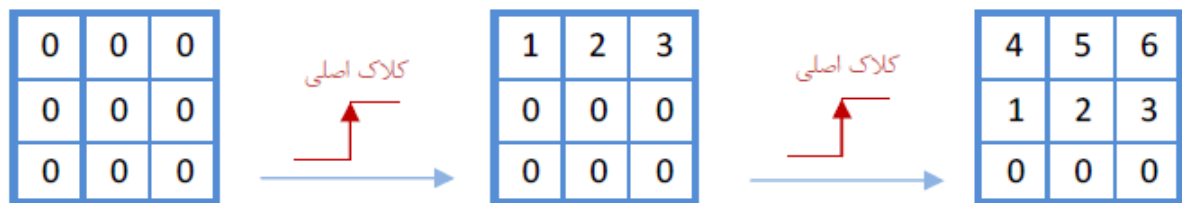


شکل 2- نحوه حرکت G_X بر روی تصویر اصلی

از آنجایی که نمی‌خواهیم این کار را به صورت آفلاین انجام دهیم، و می‌خواهیم در هر لحظه مشخص کنیم که هر پیکسل آیا لبه هست یا خیر، به صورت زیر عمل می‌کنیم:

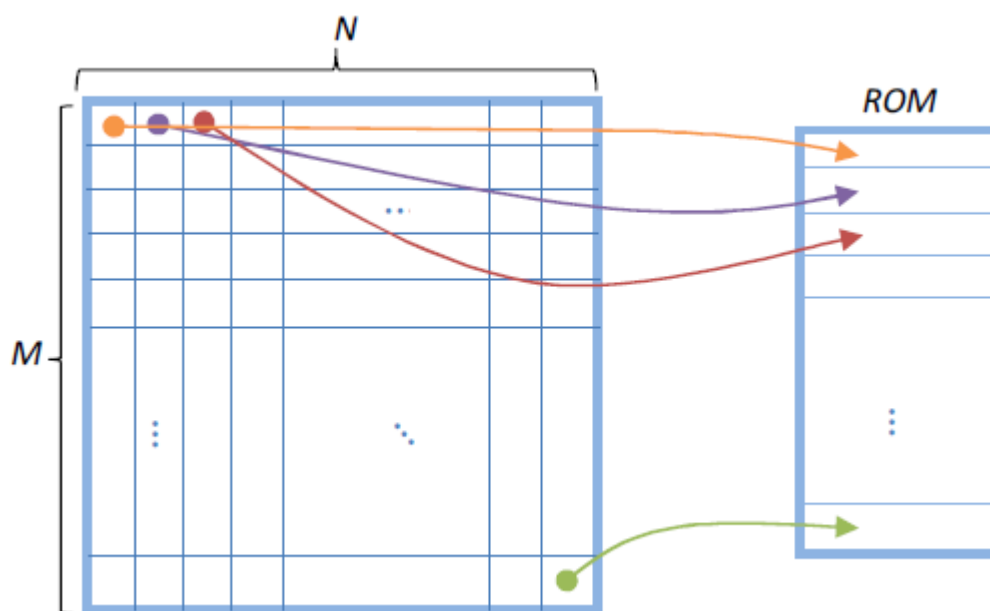
1. ابتدا کلاکی سه برابر کلاک اصلی تولید می‌کنیم به گونه‌ای که در هر لبه از این کلاک تولید شده بتوانیم یک خانه از حافظه را خوانده و مقدار آن را ذخیره کنیم. بدین ترتیب در هر لبه از کلاک اصلی محتویات سه خانه از حافظه در دسترس است.
2. این مقادیر خوانده شده را در یک سطر از حافظه 3×3 به گونه‌ای ذخیره کنید که بتوان در کلاک اصلی بعدی مقادیر جدید خوانده شده را به این حافظه اضاف نمود. بدین ترتیب می‌توانیم یک حافظه 3×3 شامل مقادیر بخشی از تصویر داشته باشیم. به عبارت دیگر می‌خواهیم در هر لحظه اطلاعات بخشی از تصویر (به صورت بلوک 3×3) در دسترس باشد.
3. اکنون با داشتن این بلوک از تصویر، عملیات دو ماتریس فوق را بر روی این بلوک اعمال کرده و ضریب وزنی پیکسل خروجی را تعیین می‌کنیم.
4. با تعیین یک سطح آستانه، رنگ مناسب برای این پیکسل را تعیین کنید. (مشخص کردن پیکسل به عنوان لبه)
5. رنگ تعیین شده را براساس کلاک اصلی (کلاکی که VGA با آن کار می‌کند)

توجه: حافظه 3×3 در نظر گرفته شده در ابتدا با صفر پر شده است و باید در هر کلاک اصلی مقادیر جدید در یک سطر جدید نوشته شوند یا مقادیر قبلی شیفت داده شده (مثلا به اندازه یک سطر) و مقادیر جدید جایگزین شوند. به شکل 3 توجه کنید:



شکل 3- نحوه قرار دادن بیت‌های خوانده شده از حافظه

توجه: اطلاعات ذخیره شده از تصویر در ROM به صورت زیر است:



شکل 4- تصویر ذخیره شده در ROM

توجه: پیشنهاد می‌شود کلاک اصلی را 25 MHz را در نظر بگیرید و کلاک 3 برابر آن را تولید کنید.

موفق باشید

