



دانشگاه علم و صنعت

دانشکده مهندسی کامپیوتر

درجه تحصیلی: کارشناسی

تکلیف 3 CAD

پرنیان شاکریان 99400064

استاد:

دکتر خدادادی

سال تحصیلی: تیر ۱۴۰۲

سوال ۳

در این کد یک سیستم `filter` را با استفاده از مفهوم عناصر پردازش (`PEs`) برای سیگنال دیجیتال پیاده سازی می کنیم. کد از دو نهاد تشکیل شده است: "`filter`" و "`PE`"، به همراه کتابخانه ها و بسته های پشتیبانی کننده.

`filter` entity " نمایانگر سیستم کلی `filter` است و دارای یک پارامتر عمومی `n` بوده که تعداد `PE` ها را در سیستم تعیین می کند. `PE` entity "`PE`" یک عنصر پردازشی واحد را نشان می دهد که ورودی را از دو آرایه میگیرد، نتیجه را محاسبه می کند و از طریق پورت `results` خروجی را می دهد.

۱. `array_no1` از نوع `filter_type` (نماینده ضرایب `filter`)

۲. `array_no2` از نوع `PE_tmp` (نماینده داده های موقت)

معماری `filter` چندین نمونه از `entity «PE»` را با استفاده از دستورات تولید ایجاد می کند. همچنین شامل سیگنال ها و انواع لازم برای مدیریت داده های ورودی، خروجی و میانی است. این معماری شامل یک تابع ناخالص به نام "`Random_Filter`" است که ضرایب `filter` تصادفی را در یک محدوده مشخص تولید می کند. این تابع از متغیرهای مشترک "`core1`" و "`core2`" برای تولید اعداد تصادفی استفاده می کند. یک تابع ناخالص دیگر، "`core`"، مقادیر متغیرهای مشترک را تنظیم می کند و یک مقدار بولی برمی گرداند. این معماری همچنین شامل فرآیندهایی برای خواندن داده های ورودی از یک فایل و تخصیص آن به سیگنال "`IFM`" است و از دسته فایل و توابع `textio` برای عملیات فایل استفاده می کند. علاوه بر این، ضرایب `filter` تصادفی تولید می کند و آنها را با استفاده از تابع "`Random_Filter`" به سیگنال "`testing_filter`" اختصاص می دهد. در نهایت، معماری از دستورات تولید تودرتو برای ایجاد یک ساختار آرایه چند بعدی استفاده می کند که سلسله مراتب نمونه های `PE` را تعریف می کند. این نمونه هایی از `PE` را بر اساس پارامتر عمومی "`n`" تولید می کند و تعداد لایه های مورد نیاز را ایجاد می کند.

کد پایتون عملیات پردازش تصویر را با استفاده از کتابخانه `OpenCV` در پایتون نشان می دهد. متغیر `img_path` مسیر فایل تصویری که باید پردازش شود را ذخیره می کند. تابع `cv2.imread` برای خواندن تصویر در حالت خاص استفاده می شود و آن را به متغیر `img` اختصاص می دهد. تابع `cv2.bitwise_not` بر روی متغیر `img` اعمال می شود و در نتیجه یک نسخه معکوس از تصویر ایجاد می شود. تصویر معکوس در متغیر `img_reverted` ذخیره می شود. تصویر `img_reverted` بر `img_reverted` تقسیم می شود تا مقادیر پیکسل بین ۰ و ۱ عادی شود. تصویر نرمال شده در متغیر `new_img` ذخیره می شود. در آخر فایلی به نام "`input_vectors`" را در حالت نوشتن باز کرده سپس در محدوده (۰) و محدوده (۲۲۵) با استفاده از حلقه های تو در تو تکرار می شود. در داخل حلقه، کد مقادیر پیکسل را از آرایه `new_img` در فایل می نویسد که با فاصله از هم جدا شده اند. در پایان هر سطر، یک کاراکتر خط جدید اضافه می شود و در نهایت، فایل با استفاده از `f.close()` بسته می شود.