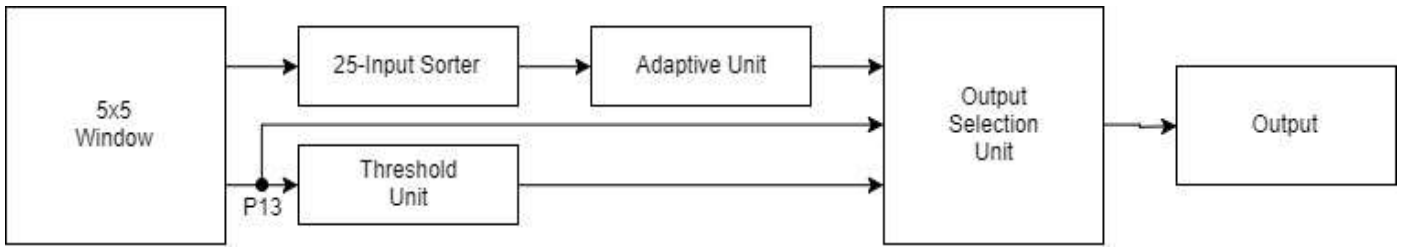
معماری قابل پیکربندی مجدد فیلتر میانه تطبیق پذیر- یک رویکرد بر اساس FPGA برای حذف نویز ضربه

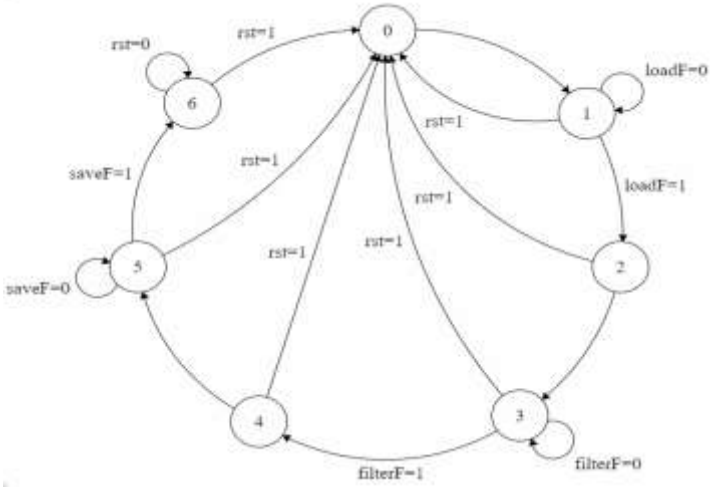


شکل ۱. بلوک دیاگرام فیلتر میانه تطبیق پذیر سوییچینگ

پیاده سازی

الف) ماژول main

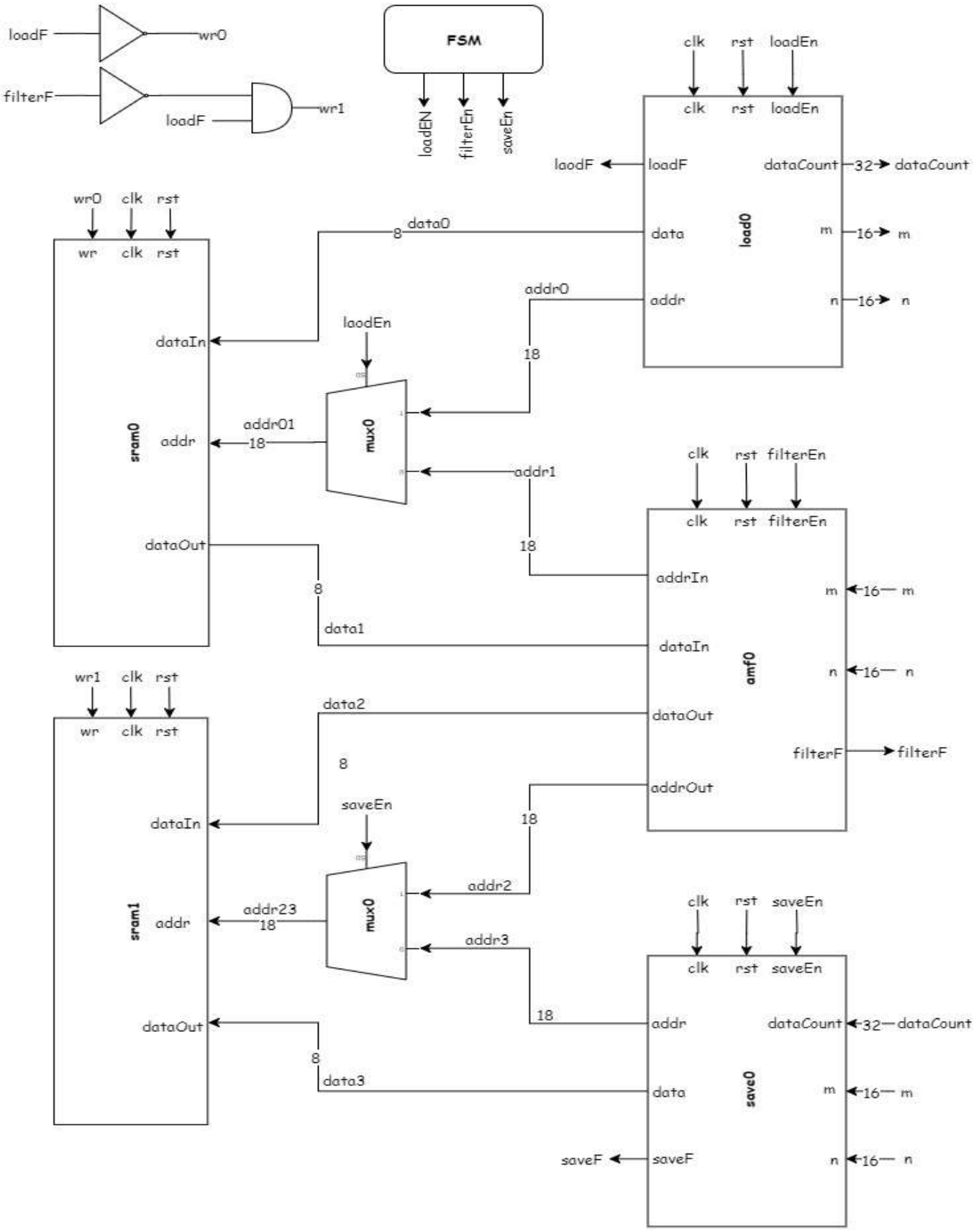
در این ماژول ابتدا داده تصویر توسط ماژول load از فایل HEX با آدرس نسبی "../inputImage.hex" خوانده شده و در حافظه sram0 ذخیره می‌گردد. سپس ماژول amf مقادیر را از sram0 خوانده، پس از انجام پردازش، مقدار هر پیکسل فیلتر شده را در حافظه sram1 ذخیره میکند. بعد از آن ماژول save داده‌ها را از sram1 خوانده و در فایل با آدرس نسبی "../outputImage.hex" به صورت HEX ذخیره می‌کند. ماژول main شامل یک FSM (شکل ۳) می‌باشد که سیگنال های کنترلی دیگر زیر ماژول‌ها را با توجه به وضعیت آنها ایجاد می‌کند.



شکل ۲. ماشین حالت متناهی ماژول main

State	Signals
0	loadEn = 1, state = 1
1	if(loadF == 1) state = 2
2	loadEn = 1, filterEn = 1, state = 3
3	If(filterF == 1) state = 4
4	saveEn = 1, filterEn = 0, state = 5
5	If(saveF == 1) saveEn = 0, state = 6
6	endF = 1

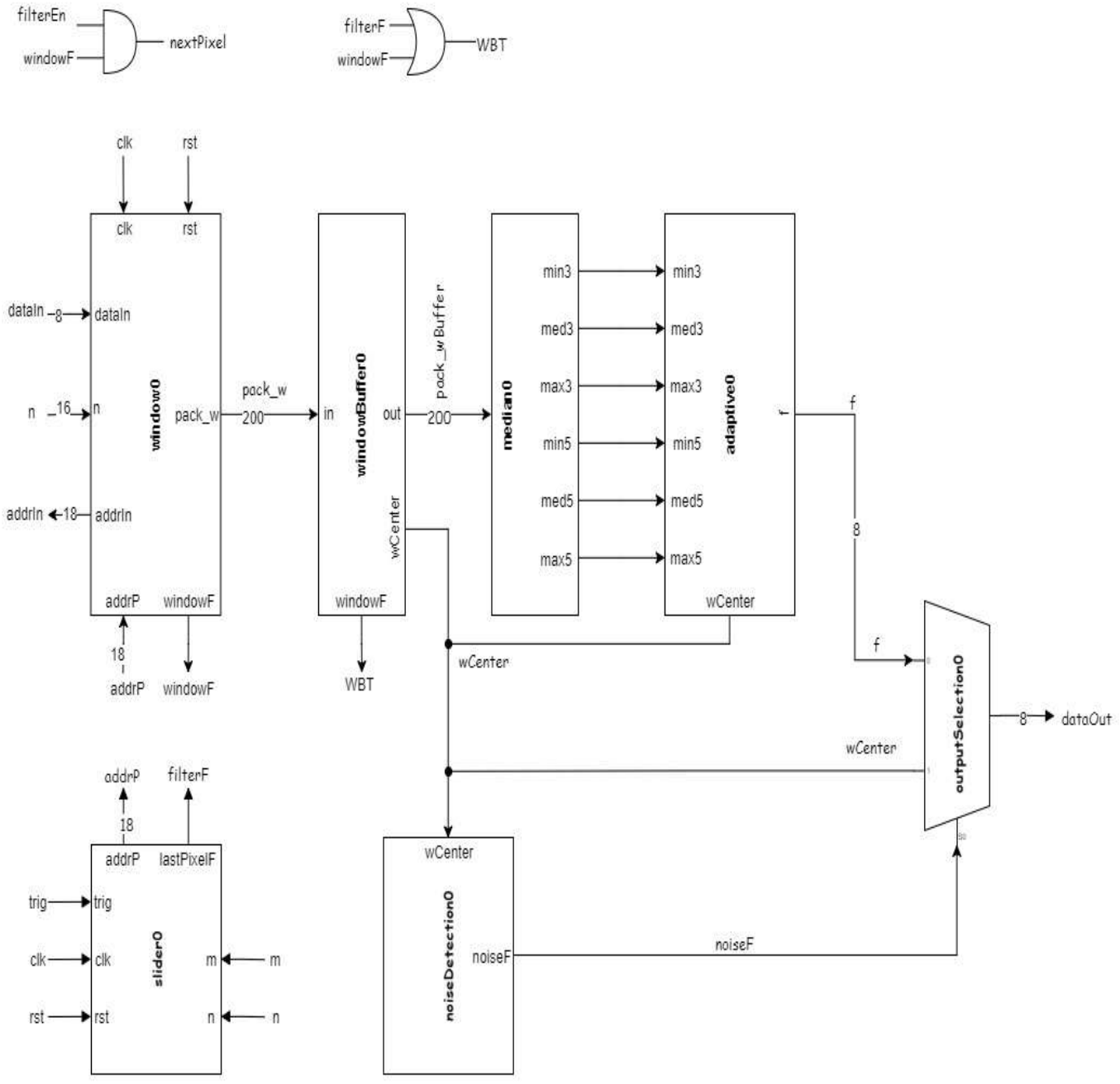
جدول ۱. حالت ماشین حالت متناهی ماژول main



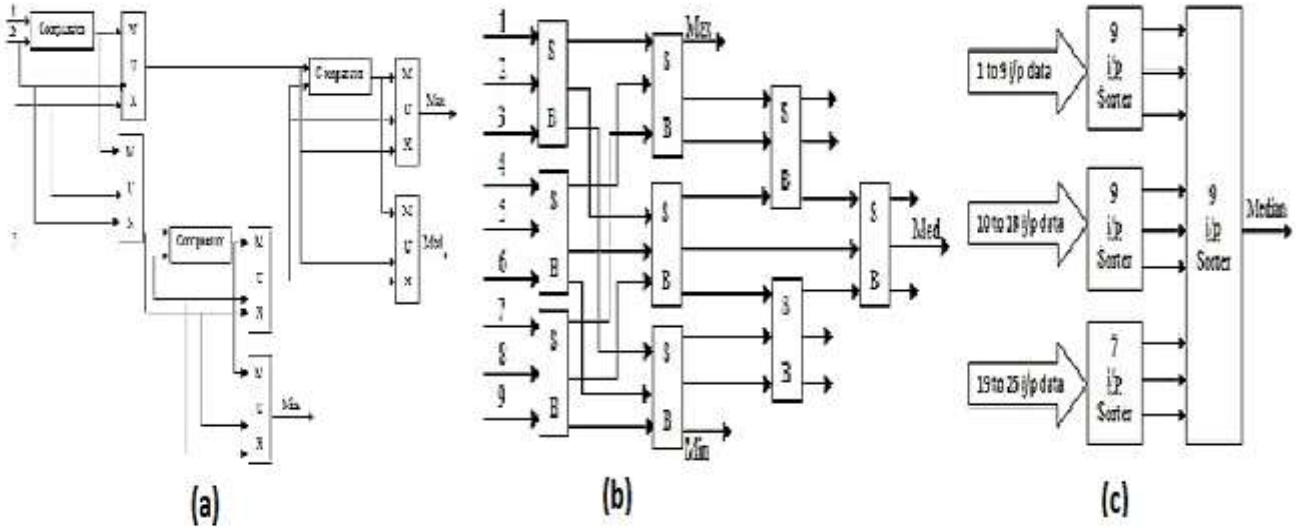
شکل ۳. بلوک دیاگرام ماژول main

ب) ماژول amf

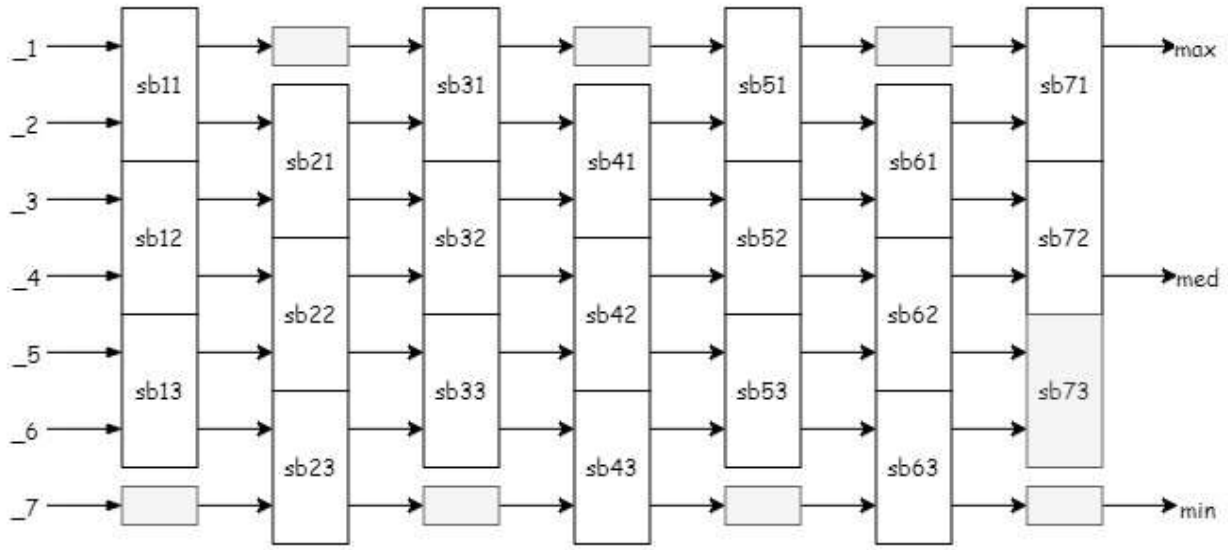
در این ماژول، توسط ماژول window یک پنجره پنج در پنج در تصویر ایجاد شده و خروجی آن به صورت یک بردار ۲۰۰ بیتی (۵*۵*۸) پس از یک مرحله بافر شدن، به ماژول median داده می‌شود. برای کاهش زمان پردازش، معماری پیشنهاد شده برای محاسبه مقدار میانه به صورت موازی طراحی شده است. این ماژول مقدار میانه، مینیمم و ماکسیمم را برای دو پنجره ۳ در ۳ و ۵ در ۵ محاسبه کرده و به ماژول adaptive می‌دهد. ماژول adaptive با توجه به مقادیر min3،med3،max3، min5 ، max5، med5 و مقدار پیکسل مرکز پنجره (wCneter)، یکی از مقادیر med5، med3 و یا wCenter را به عنوان خروجی انتخاب می‌کند. برای جلوگیری از تغییر پیکسل های غیر نویزی، ماژول noiseDetection مقدار پیکسل مرکز پنجره را بررسی کرده، اگر مابین دو عدد از پیش تعیین شده بود، پیکسل غیر نویزی تشخیص داده شده و خروجی ماژول outputSelection از مقدار خروجی ماژول adaptive (سیگنال f) به مقدار wCenter تغییر می‌کند.



شکل ۴. بلوک دیاگرام ماژول amf



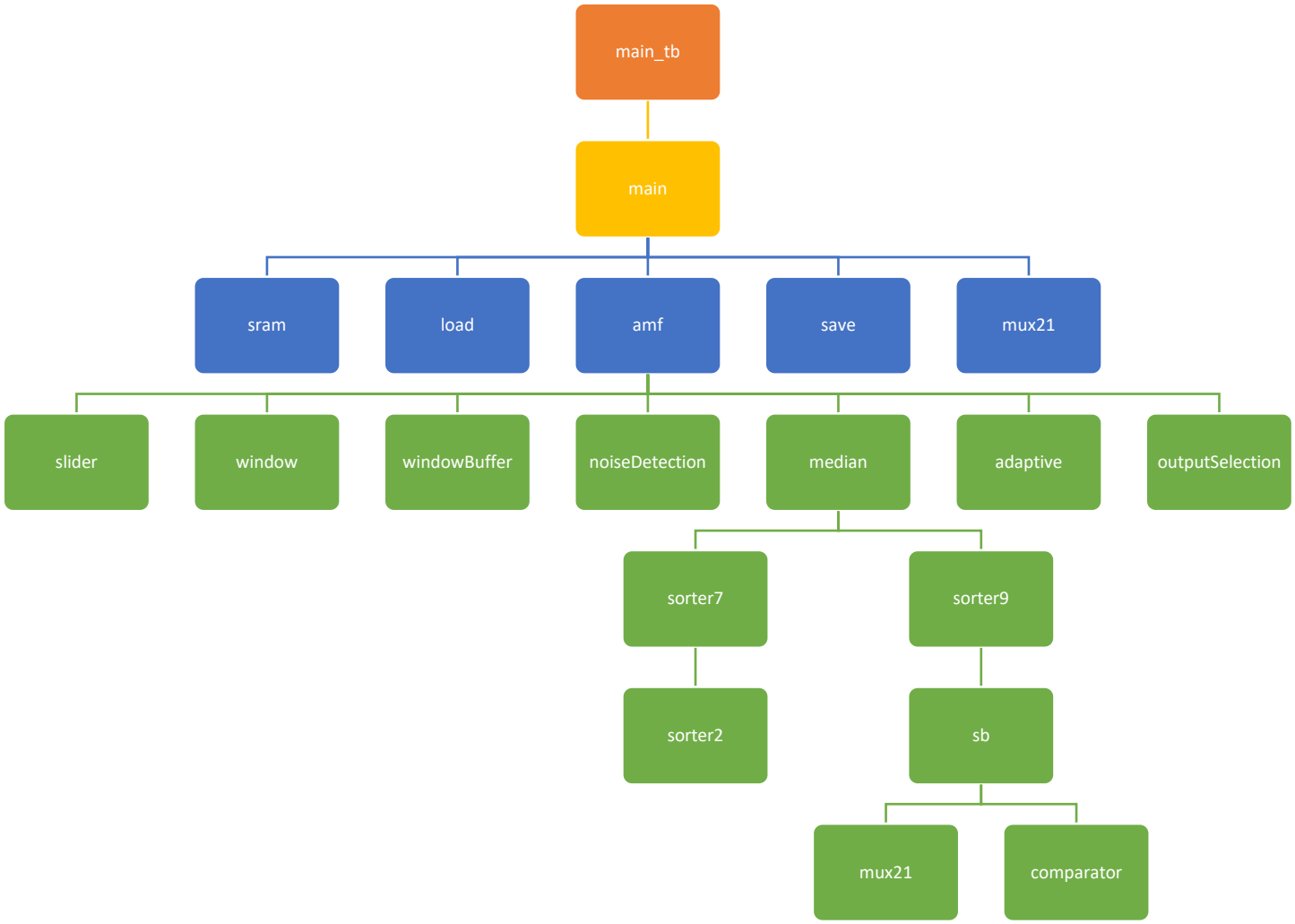
شکل ۵. بلوک دیاگرام ماژول مرتب کننده ۳ ورودی (a)، ۹ ورودی (b) و ۲۵ ورودی



شکل ۶. بلوک دیاگرام ماژول مرتب کننده ۷ ورودی به روش حبابی

```
// Select Window Size (ws = 1 : 5x5, ws = 0 : 3x3)
assign ws = ((min3 == med3)) || (med3 == max3) ? 1 : 0;
// 3x3 Window
assign out3 = ((wCenter == min3) || (wCenter == max3)) ? med3 : wCenter;
// 5x5 Window
assign out5 = ((wCenter == min5) || (wCenter == max5)) ? med5 : wCenter;
// Output Selection
assign out = ws ? out5 : out3;
```

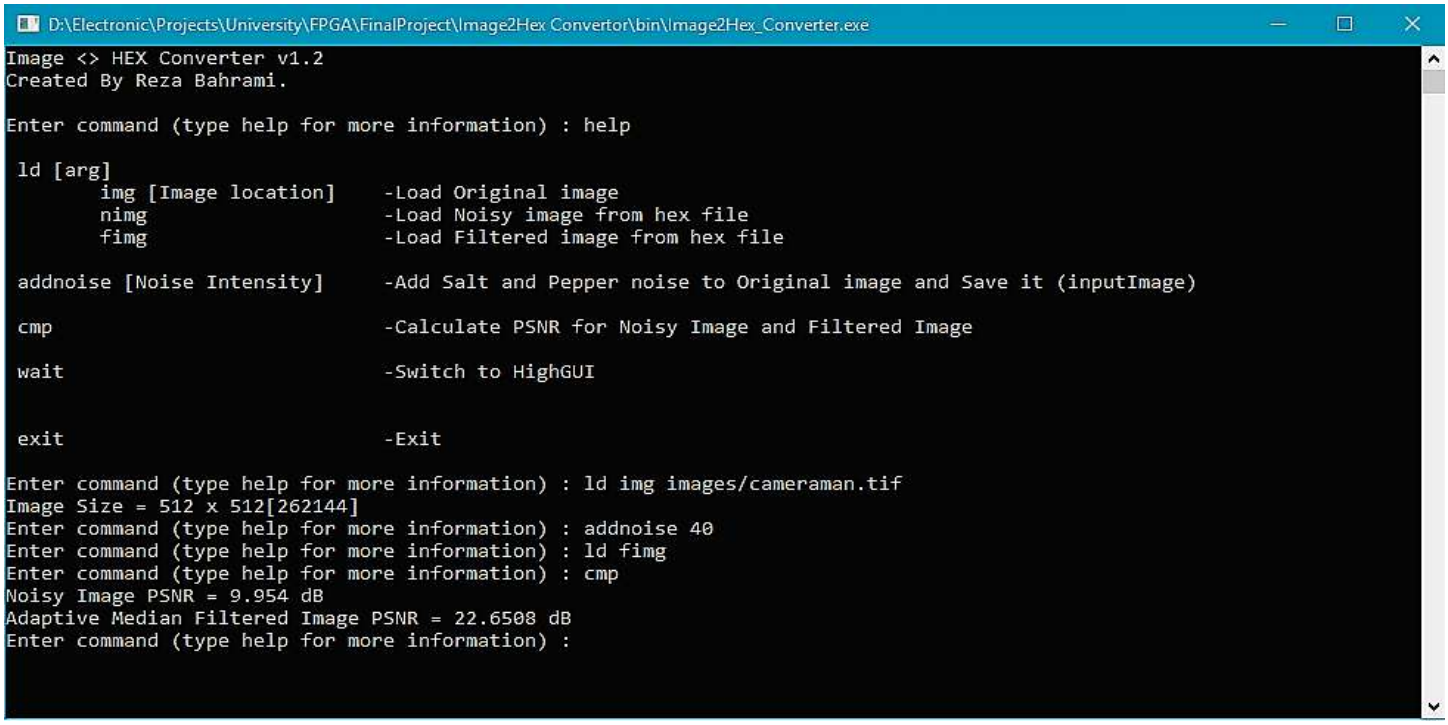
شکل ۷. قسمتی از ماژول adaptive



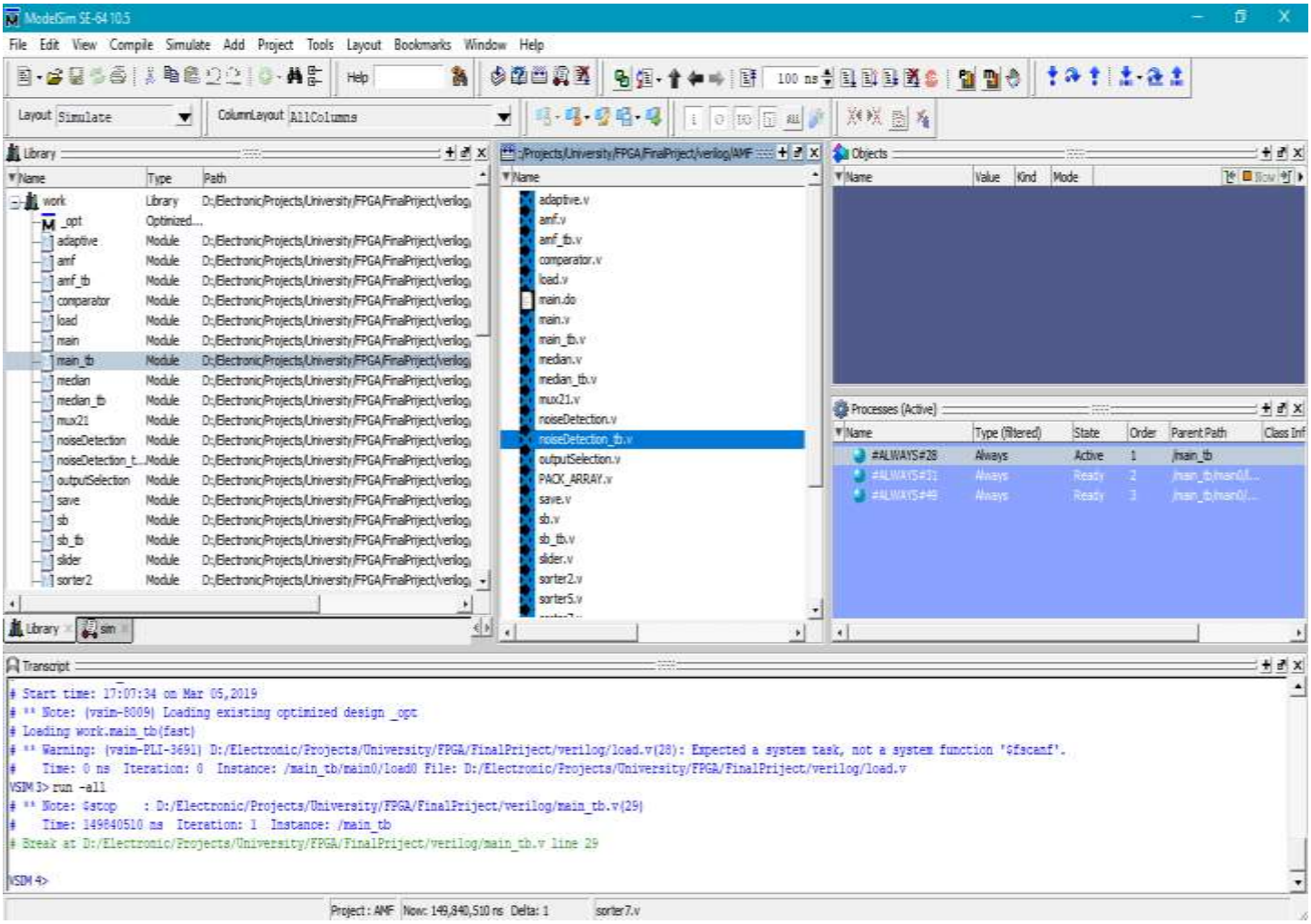
شکل ۹. نمودار سلسله مراتب ماژول ها

نحوه انجام کار

ابتدا توسط نرم افزار image2Hex Converter (Image2Hex Convertor\bin) یک تصویر را باز کرده (دستور ld img [file location]) و به آن نویز نمک فلفلی اضافه کرده و به صورت داده خام در فایل inputImage.hex ذخیره می‌کنیم (دستور addnoise [noise intensity]). سپس در نرم افزار ModelSim پس از کامپایل تمامی ماژول ها، کتابخانه main_tb در شاخه work را شبیه سازی کرده (دستور vsim work.main_tb) و با دستور run –all شبیه سازی را اجرا می کنیم. پس از پایان شبیه سازی، فایل outputImage.hex که توسط انجام شبیه سازی ایجاد شده است را توسط نرم افزار image2Hex Convertor باز کرده و نمایش می‌دهیم (دستور ld fimg). برای محاسبه و مقایسه مقدار PSNR برای تصویر نویزی و فیلتر شده، از دستور cmp استفاده می‌کنیم.



شکل ۸. نرم افزار Image2Hex Convertor



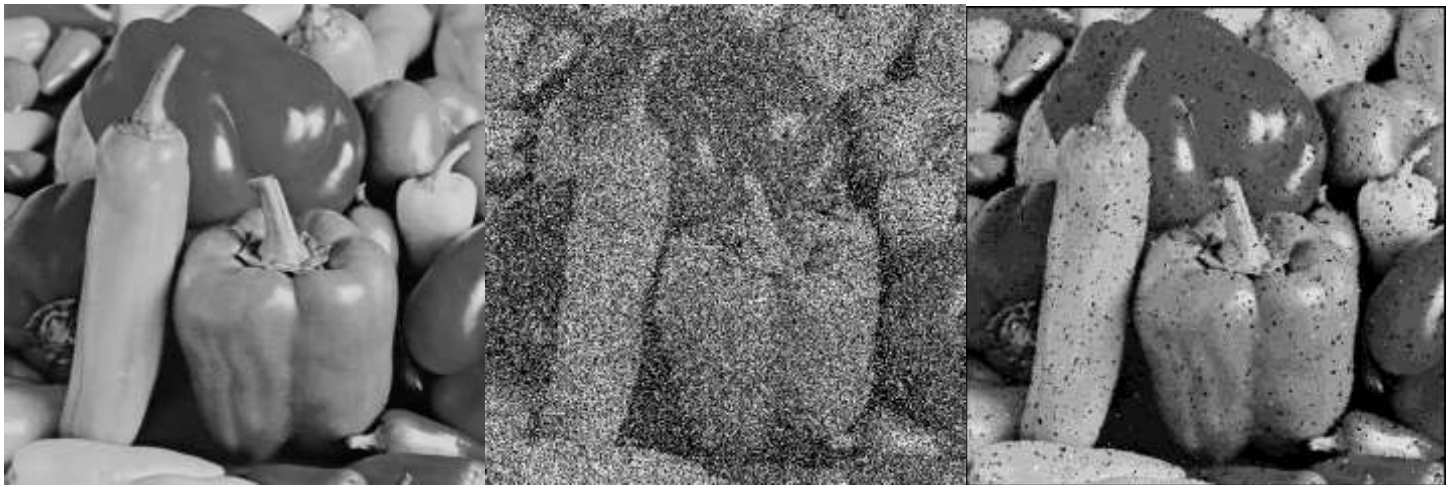
شکل ۹. تصویری از محیط نرم افزار ModelSim پس از اجرای شبیه سازی

۵ – نتایج

در شکل‌های ۱۲، ۱۳ و ۱۴، نتایج اعمال فیلتر پیشنهادی بر تصاویر Lena، Cameraman و Peppers نشان داده شده است. همچنین در جدول ۲ مقدار PSNR برای این تصاویر در حضور نویز نمک فلفلی با چگالی‌های متفاوت ارائه شده است.

شکل ۱۲. به ترتیب از چپ به راست: تصویر اصل، تصویر با نویز نمک فلفلی با چگالی نویز ٪۲۰ ، تصویر فیلتر شده

شکل ۱۳. به ترتیب از چپ به راست: تصویر اصل، تصویر با نویز نمک فلفلی با چگالی نویز ٪۴۰ ، تصویر فیلتر شده



شکل ۱۴. به ترتیب از چپ به راست: تصویر اصل، تصویر با نویز نمک فلفلی با چگالی نویز ٪۸۰ ، تصویر فیلتر شده

Image	Noise Intensity	20%	40%	60%	80%	90%
Lena	Noisy Image PSNR (dB)	12.8952	10.308	8.95303	8.05577	7.7247
	Filtered Image PSNR (dB)	20.8468	2.5107	19.5736	17.3182	15.9374
Cameraman	Noisy Image PSNR (dB)	12.5089	9.954	8.59378	7.77444	7.43354
	Filtered Image PSNR (dB)	22.9719	22.6508	21.2431	18.3575	16.5616
Peppers	Noisy Image PSNR (dB)	12.7234	10.1474	8.83952	7.98839	7.66615
	Filtered Image PSNR (dB)	22.8636	22.5413	21.4205	18.6235	16.8495

جدول ۲. نتایج فیلتر میانه تطبیق پذیر بر روی Lena، Cameraman و Peppers بر حسب PSNR

۷ – نرم افزار

برای انجام و ارزیابی این پروژه از نرم افزارهای ذیل استفاده شده است.

a. Mentor Graphics ModelSim SE 10.5: کامپایل و شبیه سازی کد توصیف سخت افزار (زبان Verilog)
b. Microsoft Visual Studio Code: ویرایش گر کد توصیف سخت افزار (زبان Verilog)
c. Microsoft Visual Studio 2017 Community: ساخت برنامه Image2Hex Convertor (زبان C++ به همراه مجموعه کتابخانه های OpenCV)
d. Draw.io و madebyevan.com/fsm: طراحی بلوک دیاگرام ها
e. Microsoft Office

منابع

[1] Mukherjee, Manali, and Mausumi Maitra. "Reconfigurable architecture of adaptive median filter—An FPGA based approach for impulse noise suppression." *Proceedings of the 2015 Third International Conference on Computer, Communication, Control and Information Technology (C3IT)*. IEEE, 2015.

[2] Rafael C. Gonzalez, and Richard E. woods, Digital Image Processing, 3rd edition, Prentice Hall,2009.

[3] Bates, Gavin L., and Saeid Nooshabadi. "FPGA implementation of a median filter." *TENCON'97 Brisbane-Australia. Proceedings of IEEE TENCON'97. IEEE Region 10 Annual Conference. Speech and Image Technologies for Computing and Telecommunications (Cat. No. 97CH36162)*. Vol. 2. IEEE, 1997.

[4] Fahmy, Suhaib A., Peter YK Cheung, and Wayne Luk. "Novel FPGA-based implementation of median and weighted median filters for image processing." *International Conference on Field Programmable Logic and Applications, 2005.* IEEE, 2005.

[5] Prasad, Devi, et al. "Sorting networks on FPGA." *Proceedings of the WSEAS International Conference on Telecommunications and Informatics (TELE-INFO)*. 2011.