

به نام خدا

## تشخیص تابلو های محدود کننده سرعت با استفاده از *FPGA*

گردآوری

مژده مهدوی، محمد امین امیری، علیرضا بهراد، نوشین مهدوی

### کلمات کلیدی

*FPGA*، تطبیق الگو، تابلو محدود کننده سرعت

### چکیده

در این مقاله یک روش سخت افزاری با سرعت بالا برای تطبیق الگوی مبتنی بر حداقل خطا جهت کاراکترهای تابلوهای محدود کننده سرعت ارائه شده است. الگوریتم پیشنهادی در قالب یک ساختار موازی پیاده سازی شده است. کلیه ماژول های این سخت افزار با استفاده از *VHDL* طراحی و توصیف شده اند. برای یک تصویر ورودی با ابعاد  $32 \times 48$ ، زمان محاسبه برابر ۵۵۰ میکرو ثانیه می باشد. *FPGA* پیشنهادی ما برای این سخت افزار، *XC3S200PQ208* می باشد. سنتز و پیاده سازی این سخت افزار با استفاده از نرم افزار *ISE* از شرکت *Xilinx* انجام شده است. نتایج شبیه سازی با استفاده از *ModelSim* بدست آمده است.



## ۱. مقدمه

بررسی مشکلات حمل و نقل نشان میدهد که اکثر تصادفات رانندگی در اثر اهمیت ندادن راننده ها به علایم راهنمایی رانندگی است. دسته بزرگی از تصادفات بدلیل بی دقتی نسبت به تابلوهای محدودکننده سرعت و خارج شدن از سرعت مجاز میباشد. امروزه برای کاهش مشکلات رانندگی استفاده از سیستمهای ترابری هوشمند (*Intelligent Transportation System*) در جریان است که در آن سطحی از هوشمندی برای خودرو تعریف میشود که بتواند به عنوان یک سیستم کمک راننده عمل هدایت اتومبیل را تسهیل کند و در مواقع اضطراری و هنگامیکه سرعت خودرو بیش از حد مجاز است به راننده هشدار داده و در مکانیزمهای خاص حتی سرعت اتومبیل را نیز محدود کند. از مشخصه های مهم این سیستمها کاربرد پردازش تصویر به صورت *real time* و سرعت بالا است. برخی از عملیاتی که توسط این سیستمها انجام می شود شامل تشخیص تابلوی محدود کننده سرعت که دوایر قرمز هستند و تشخیص کاراکتر درون این علایم است. یکی از روشهایی که برای تفسیر الگو مورد استفاده قرار می گیرد تطبیق الگو (*Template Matching*) است. تطبیق الگو یک روش علمی است که هدفش کلاسه بندی اشیا به تعدادی از کلاسهها میباشد [1]. تابلوهای محدود کننده سرعت شامل یازده عدد است (۱۰ و ۲۰ و ۳۰ و ۴۰ و ۵۰ و ۶۰ و ۷۰ و ۸۰ و ۹۰ و ۱۰۰ و ۱۱۰) که کلاسه بندی آنها برای تشخیص تابلوهای راهنمایی رانندگی ضروری میباشد. اقداماتی که تا کنون برای تفسیر کاراکتر تابلوهای محدود کننده سرعت صورت گرفته مبتنی بر نرم افزار میباشد. همچنین شیوه هایی که معمولاً استفاده میشوند روشهای شبکه عصبی و تطبیق الگو است. در مرجع [2] برخی از الگوریتم های شبکه عصبی مانند *MLP, LVQ, Hopfield, RBF* مورد بررسی و مقایسه قرار گرفته است و بهترین روش نسبت به بقیه موارد الگوریتم *MLP* معرفی شده است. باید در نظر داشت که ورودی تصویر در مرجع [2] به ابعاد  $30 \times 30$  نرمالیزه شده است و زمان مصرفی آن ۷۷ میلی ثانیه بوده است. مرجع [3] بیان می کند که برای تصویر با ابعاد  $30 \times 30$  تاخیر مورد انتظار برای پیاده سازی شبکه های عصبی روی پردازنده های *DSP* برابر ۳۰ تا ۴۰ میلی ثانیه میباشد.

مرجع [4] تطبیق الگو را برای مقاصد کلی پیاده سازی نمود و تصویر ورودی آن دارای ابعاد  $512 \times 512$  پیکسل می باشد. تاخیر محاسباتی آن سخت افزار با مصرف ۱۴۳۳۴ عدد از *Slice* های *XC2V3000*, برابر ۱۴.۸ میلی ثانیه می باشد.

الگوریتم پیشنهادی ما در قالب یک ساختار موازی پیاده سازی شده است. کلیه ماژول های این سخت افزار با استفاده از *VHDL* طراحی و توصیف شده اند. برای یک تصویر ورودی با ابعاد  $48 \times 32$  زمان

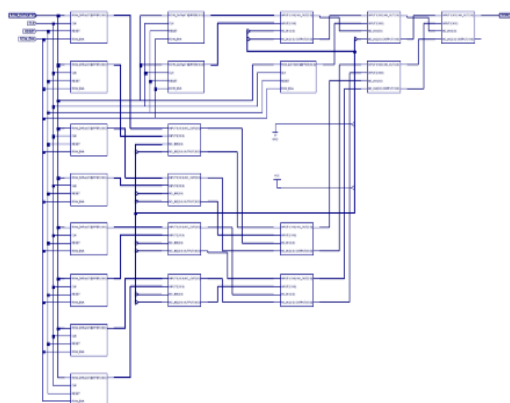
محاسبه برابر ۵۵۰ میکرو ثانیه می باشد. *FPGA* پیشنهادی ما برای این سخت افزار, *XC3S200PQ208* میباشد که تعداد ۱۲۸۹ عدد از *Slice* های آن مصرف شده اند. در ادامه و در بخش II, توضیح کوتاهی از الگوریتم تطبیق الگو را ارائه می دهیم. در بخش III ساختار موازی پیشنهادی و مراحل عملیات تشریح می شوند. برخی از نتایج شبیه سازی در بخش IV ارائه می شوند. در بخش V نیز نتیجه گیری انجام شده است.

## ۲. الگوریتم تطبیق الگو

الگوریتم انتخابی ما برای تشخیص کاراکترها *Template Matching* با استفاده از حداقل خطا است. در این روش الگوها در اندازه یکسان در حافظه *ROM* ذخیره شده اند. تصویری که قرار است بررسی شود باید در ابتدا به اندازه الگوها نرمالیزه شود. در مرحله بعد اختلاف تصویر ورودی با تک تک الگوها اندازه گیری میشود. با مقایسه این اختلافها هر الگویی که دارای کمترین اختلاف با تصویر ورودی بود به عنوان مشابه ترین نمونه انتخاب میشود و تصویر ورودی معادل الگوی انتخاب شده متناظر با آن میباشد.

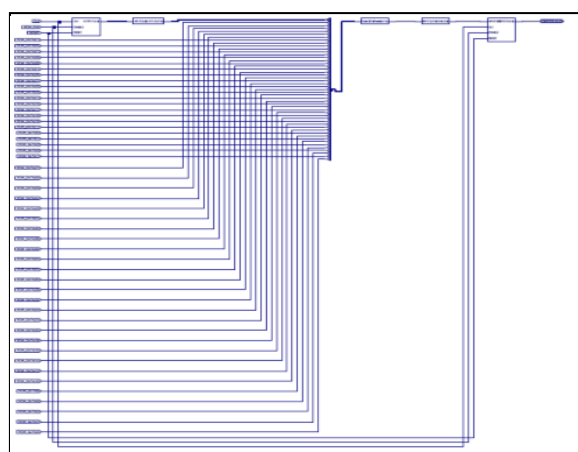
## ۳. ساختار سخت افزار

در این بخش ما ساختار پیشنهادی برای پیاده سازی الگوریتم تطبیق الگو مبتنی بر حداقل خطا را تشریح میکنیم. ساختار پیشنهادی ما, یک ساختار موازی می باشد و تصویر ورودی بطور موازی به ماژول های محاسبه گر اختلاف اعمال می شوند. تصویر ورودی این سخت افزار باید باینری بوده و دارای ابعاد  $48 \times 32$  باشد. تصویر ورودی بصورت سطر به سطر وارد می شود و یک ورودی فعال ساز نشان می دهد که سطر ورودی معتبر می باشد. سیگنال های کلاک و ریست, دیگر ورودی های این ماژول اصلی می باشند. خروجی این ماژول ۴ بیتی بوده و اعداد ۱, ۲, ..., ۱۱ را متناسب با کاراکترهای ۱۰, ۲۰, ... و ۱۱۰ اختیار می کند. همانطور که در شکل ۱ می بینید, ۱۱ ماژول محاسبه گر اختلاف, اختلاف بین تصویر ورودی و الگوهای ذخیره شده در بلوک های حافظه شان را می یابند و سپس ماژول های مقایسه گر, اختلاف ها را با هم مقایسه کرده و حداقل اختلاف را بدست آورده و الگوی مربوطه را انتخاب می نمایند. ماژول های محاسبه گر اختلاف شامل یک حافظه, یک تولید کننده آدرس برای حافظه, یک جمع کننده و یک انبار می باشند.



شکل ۱: ساختار سخت افزار

شکل ۲ ساختار داخلی یک ماژول های محاسبه گر اختلاف را نشان می دهد. تصویر ورودی بصورت سطر به سطر به این ماژول ها اعمال می شود. در ابتدا، سیگنال ریست باید بمدت یک پریود کلاک برابر یک باشد. پس از آن، برای هر سطر معتبر، ورودی فعال ساز سطر باید برابر یک شود. بازای هر سطر ورودی، سطر متناظر از حافظه با آن  $xor$  می شود و ۴۸ بیت خروجی بدست می آید.



شکل ۲: ماژول محاسبه گر اختلاف

در این ۴۸ بیتی، صفرها بیان کننده پیکسل های مشابه و یک ها بیان کننده اختلاف می باشند. بنابراین با جمع کردن این یک ها می توان اختلاف سطر ورودی و سطر متناظر در الگو را بدست آورد. این عملیات برای تمام ۳۲ سطر انجام شده و نتیجه در انباره جمع می شود. خروجی انباره برابر اختلاف تصویر و الگوی خاص آن ماژول می باشد.

یک پریود کلاک پس از اینکه آخرین سطر اعمال شده و جهت بدست آوردن اختلاف استفاده گردید، خروجی علامت معتبر می باشد. برای اعمال تصویر بعدی، سیگنال ریست باید برای یک پریود کلاک برابر یک شود. پس از آن می توان تصویر را مطابق آنچه بیان شد، اعمال نمود.

کلیه ماژول های این سخت افزار با استفاده از *VHDL* طراحی و توصیف شده اند. *FPGA* پیشنهادی ما برای این سخت افزار، *XC3S200PQ208* می باشد. سنتز و پیاده سازی این سخت افزار با استفاده از نرم افزار *ISE* از شرکت *Xilinx* انجام شده است. نتایج سنتز در جدول ۱ موجود می باشد.

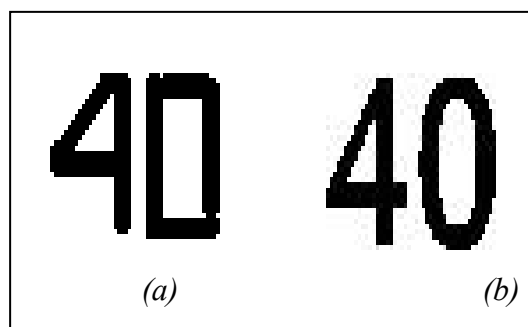
پس از مراحل *Place & Route* سخت افزار، حداکثر تاخیر گزارش شده توسط نرم افزار برابر ۱۵.۹۱۹ نانو ثانیه می باشد. این بیان می کند که فرکانس کلاک این سخت افزار حدود ۶۲MHz می باشد. از آنجائی که الگوریتم در حداقل ۳۴ کلاک کامل می شود، کل زمان محاسبات برای این الگوریتم برابر ۵۵۰ میکرو ثانیه می باشد.

جدول ۱ : نتایج سنتز

#### ۴. نتایج شبیه سازی

سخت افزار پیشنهادی با استفاده از نرم افزار *ModelSim* شبیه سازی شده است. نتایج شبیه سازی بطور کامل با انتظارات ما مطابقت دارد. تصویر الگو برای تابلو محدود کننده سرعت "۴۰" و یک ورودی معوج نوعی از همین تابلو در شکل ۳ نشان داده شده اند. نتایج شبیه سازی برای تابلو معوج نیز در شکل ۴ نشان داده شده است.

	Use d	Total	Percentag e
Slices	1289	1920	67%
Slice FlipFlops	381	3840	9%
4 input LUTs	2471	3840	64%
Bonded IOBs	55	141	39%
GCLKs	1	8	12%



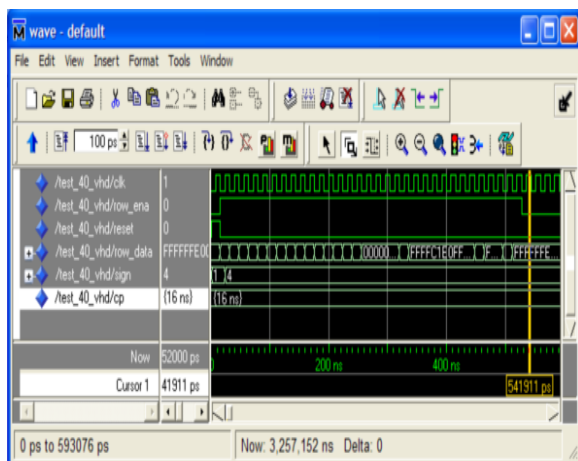
شکل ۳: (a) تصویر معوج، (b) تصویر

همانطور که در شکل ۴ می بینید، سخت افزار شبیه سازی شده توانست تابلو معوج "۴۰" را تشخیص داده و خروجی "۴" را برگرداند.

## ۵. نتیجه گیری

الگوریتم پیشنهادی ما در قالب یک ساختار موازی پیاده سازی شده است. کلیه ماژول های این سخت افزار با استفاده از VHDL طراحی و توصیف شده اند. پس از مراحل Place & Route سخت افزار، حداکثر تاخیر گزارش شده توسط نرم افزار برابر ۱۵.۹۱۹ نانو ثانیه می باشد. این بیان می کند که فرکانس کلاک این سخت افزار حدود 62MHz می باشد. از آنجائی که الگوریتم در حداقل ۳۴ کلاک کامل می شود، کل زمان محاسبات برای یک تصویر ورودی با ابعاد ۴۸\*۳۲، برابر ۵۵۰ میکرو ثانیه می باشد. FPGA پیشنهادی ما برای این سخت افزار، XC3S200PQ208 می باشد که تعداد ۱۲۸۹ عدد از Slice های آن مصرف شده اند.

با مقایسه تاخیر محاسباتی کل سخت افزار پیشنهادی ما و کارهای قبلی در می یابیم که سرعت تشخیص الگوریتم و سخت افزار پیشنهادی ما بسیار بهتر می باشد.



مراجع

شکل ۴: نتایج شبیه

S. Theodoridis, K. Koutroumbas, "PATTERN RECOGNITION", ELSEVIER Academic Press, 2003.

Y. Baleghi D., K. Mohammadi, "Speed Limit Traffic Sign Detection & Recognition", IEEE Conf. on Cybernetics and Intelligent Systems, 2004.

A. Escalera, L. E. Moreno, M. A. Salichs, J. M. Armingol, "Road Traffic Sign Detection and Classification", IEEE, 1997.

*S. Hezel, A. Kugel, R. Manner, D.M. Gavrilu, "FPGA-based Template Matching using Distance Transforms", IEEE Symposium on Field-Programmable Custom Computing Machines, 2002*