پیاده‌سازی الگوریتم minimax بر روی FPGA

امیدرضا گندمی *استاد راهنما:* دکتر بیژن علیزاده

دانشکده مهندسی برق و کامپیوتر، دانشگاه تهران

مقدمه

با پیدایش رایانه و شکل‌گیری علم هوش‌مصنوعی همواره یکی از موضوعات مورد علاقه محققان این حوزه بازی‌ها بوده‌اند، دلیل این امر می‌تواند وجود قوانین مشخص و ثابت در بازی‌ها باشد که در خیلی از مسایل دیگر وجود ندارد. الگوریتم‌های تصمیم‌گیری متفاوتی به منظور حل بازی های دو نفره ارایه شده است، الگوریتم minimax یکی از ساده ترین و راهگشاترین آنها در زمینه بازی‌های دو نفره بدون دخالت شانس است. ایده اصلی minimax برای بردن بازی انجام پرامتیازترین حرکت نیست بلکه به دنبال حرکتی است که منجر به کم امتیازترین حرکت حریف در نوبت‌های بعد شود. در این رساله طراحی یک الگوریتم تصمیم گیری minimax برای بازی paper soccer و پیاده سازی آن بر روی یک سیستم نهفته دنبال شده است.

الگوریتم پیشنهادی برای انتخاب یک حرکت از حرکت‌های ممکن، بهترین حرکت حریف در نوبت بعدی را جستجو کرده و سپس حرکتی را انتخاب می‌کند که به کم ارزش‌ترین حرکت حریف در نوبت بعد منجر شود. ملاک ارزش‌گذاری یک حرکت، فاصله نهایی توپ تا دروازه یا قابلیت سد کردن راه حریف است.

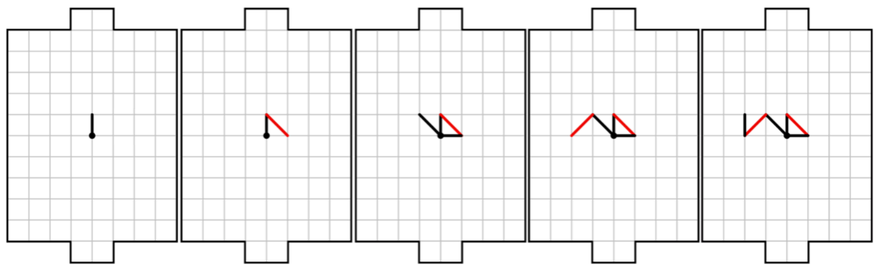
در نهایت به منظور پیاده سازی سخت افزاریِ الگوریتم بر روی FPGA، مسیرداده و کنترل‌کننده آن در سطح RTL طراحی شده و برای توصیف سخت افزاری آن از زبان systemverilog بهره گرفته شده است. برد نهایی (Altera DE-2) می‌تواند به وسیله ارتباط سریال با یک برد دیگر یا رایانه به بازی بپردازد.

قوانین بازی:

در این بازی که به طور سنتی با کاغذ و مداد انجام میشود، صفحه بازی با شبکه های ۱۰ در ۸ از مربع ها به عنوان زمین فوتبال و دو مربع در دو طرف کوتاه زمین به عنوان دروازه دو تیم در نظر گرفته می شود.

در هنگام شروع بازی، توپ فوتبال با یک نقطه در وسط صفحه بازی نشان داده می شود. هر بازیکن به نوبت با رسم یک خط توپ را از مکان فعلی به مکان جدید منتقل می کند. هر حرکت باید به نقطه مجاور در راستای عمودی، افقی یا مورب انجام شود. هیچ حرکتی نمی تواند توپ را از زمین بازی خارج کند. همچنین حرکت از روی خطی که در حال حاضر کشیده شده است مجاز نیست. بازیکنی که بازی را شروع می کند به صورت رندم انتخاب می شود.

اگر توپ به نقطه ای برسد که یک یا چند خط به آن متصل است (شامل خطوط محیطی زمین بازی)، از آن نقطه می پرد و بازیکنی که آن حرکت را انجام داده است باید به حرکت خود ادامه دهد. زمانی حرکت بازیکن متوقف می شود که توپ به نقطه ای برسد که خطی به آن متصل نیست (مگر آنکه توپ وارد دروازه شود که در این صورت یک دور بازی خاتمه می یابد). شکل حرکت های ابتدایی از یک بازی نمونه را نشان می دهد.



برنده بازی کسی را به داخل دروازه حریف برساند. اگر بازیکنی به نقطه ای برسد که باید پرش انجام دهد و نتاست که بتواند توپ واند این کار را انجام دهد بازنده خواهد بود (بسته به قوانین از پیش توافق شده ممکن است این حالت به عنوان تساوی تلقی شود).

ساختار پیشنهادی

الگوریتم در حالت کلی از سه بخشِ تشکیل درخت، ارزش گذاری حرکت ها و جستجو برای بهترین حرکت تشکیل شده است. الگوریتم پیشنهاد شده در اینجا تنها یک نوبت حرکت از طرف مقابل را بررسی می‌کند بنابراین درخت حاصله عمقی برابر سه دارد. در این بازی نیازی به ذخیره کامل درخت نیست و مسیر مورد بررسی در صورت بهتر بودن نسبت به بهترین حالت ذخیره شده قبلی جایگزین آن می‌شود و این روند تا انتها پیش می‌رود به عبارت دیگر مرحله تشکیل درخت و مرحله جستجوی درخت در یک دیگر ادغام شده‌اند. معیار ارزش گذاری حرکت‌ها خیلی ساده تنها فاصله نهایی توپ تا دروازه حریف در نظر گرفته شده است.

عمق درخت و دقیق بودن معیار ارزش‌گذاری، در ساده ترین حالت ممکن در نظر گرفته شده‌اند که باعث ساده شدن فرایند طراحی و زمان آن شده است ولی این دلیل بر ضعیف بودن هوش مصنوعی حاصل نیست و با توجه به ماهیت ساده بازی عملکرد مناسبی از خود به نمایش می‌گذارد. شبیه‌سازی کد نرم‌افزاری مؤید این مدعاست.

شالوده الگوریتم دو حلقه تو در تو است که حلقه بیرونی برای پیمایش حرکت پیشنهادی ما (moveDir) و حلقه داخلی برای پیمایش حرکت احتمالی حریف در نوبت بعدی (oppMoveDir) است.

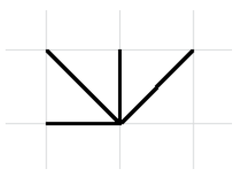
در حلقه داخلی ما به دنبال بهترین حرکت احتمالی حریف هستیم (قسمت max الگوریتم). معیار برای تشخیص بهتر بودن، فاصله نهایی توپ از دروازه خود حریف است (این معادل فاصله کمتر تا دروازه ما است). هر چه این فاصله بیشتر باشد حرکت امتیاز بیشتری دارد. در حلقه بیرونی به دنبال حرکتی هستیم که حریف در نوبت بعد مجبور به انتخاب کم امتیازترین "بهترین حرکت در نوبت بعد" شود. (قسمت min الگوریتم)

در صورت برخورد به یک دیوار یا عبور از نقطه‌ای که خطی به آن متصل است، حرکت را در یک پشته ذخیره می‌کنیم و سپس حرکت های محتمل در نقطه جدید را بررسی می‌کنیم. در پایان بررسی تمام حرکت ها از نقطه جدید، حرکت قبلی را از پشته فراخوانی می‌کنیم و به بررسی حرکت بعدی طبق روال معمول می‌پردازیم. (این روش مشابه توابع recursive در نرم‌افزار است.)

در پیاده‌سازی های نرم افزاری به دلیل عدم وجود دغدغه حافظه ممکن است برای بررسی هر حالت یک کپی از کل زمین تهیه و بر روی آن پردازش کرده و سپس آن را حذف کنند. در اینجا به دلیل لزوم بهینه سازی از لحاظ حافظه و عدم امکان تخصیص دینامیکی حافظه، برای پردازش زمین بازی تغییرات را بر روی خود زمین اعمال کرده و آن ها را در یک پشته نیز ذخیره می‌کنیم و پس از اتمام پردازش مربوطه مطابق پشته، تغییرات را بر می‌گردانیم.

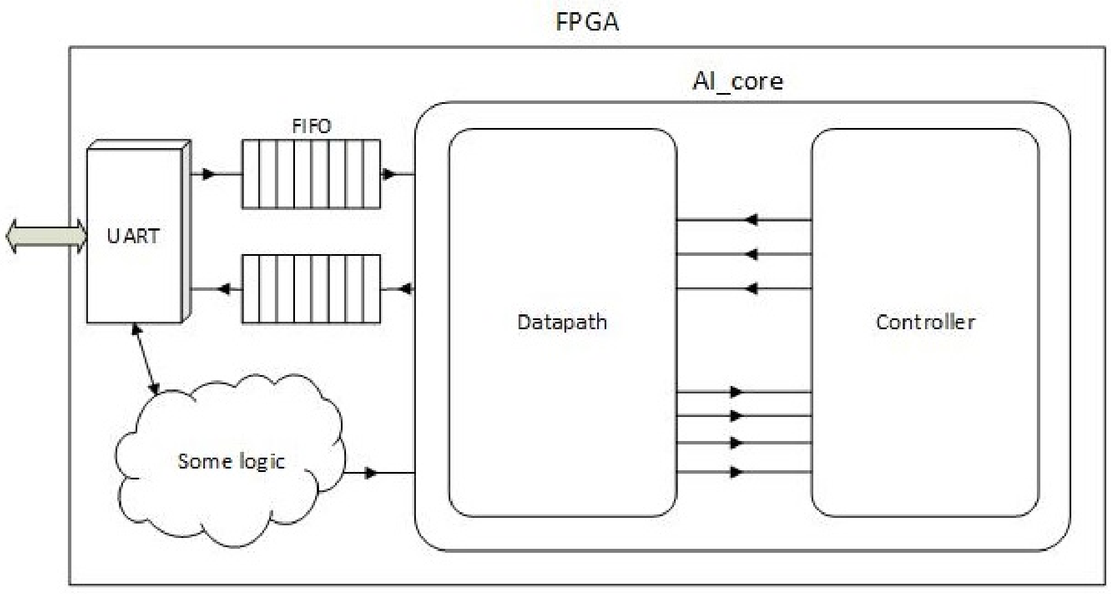
ذخیره زمین بازی:

به منظور ذخیره زمین بازی از یک آرایه دوبعدی از گره ها استفاده شده است. هر گره خود ذخیره کننده سه ویژگی است؛ نخست مشخص کننده وجود یا عدم وجود چهار خط متصل به آن و دوم این که آیا خطی به گره متصل است یا نه و سوم آیا این گره مربوط به دروازه(گل) است.



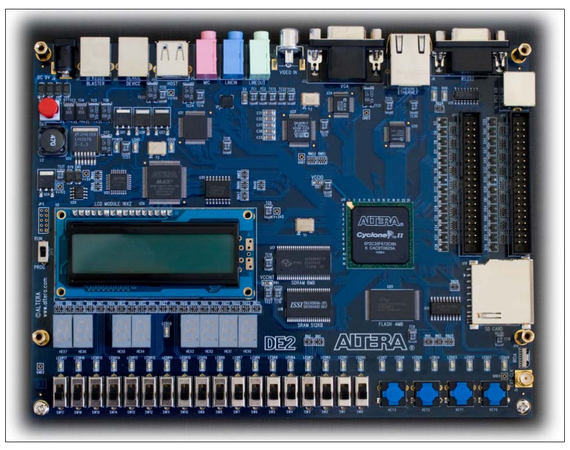
معماری سخت‌افزار:

معماری کلی ماژول‌ها در شکل زیر نشان داده شده است. مطابق تصویر هسته اصلی خود شامل مسیرداده و کنترل‌کننده است. یکwrapper برای هسته اصلی در نظر گرفته شده که وظیفه ارتباط با خارج از FPGA، مدیریت ارتباط سریال، ترجمه دستورات داور خارجی و حفظ نوبت بازی و دادن دستور متناسب به هسته اصلی را برعهده دارد.



جمع بندی

در این پروژه در گام نخست ما به انتخاب الگوریتم مناسب برای بازی مورد نظر پرداختیم، پیش نیاز این امر آشنایی اولیه با الگوریتم های تصمیم‌گیری بود. پس از شناخت طبیعت بازی و طراحی الگوریتم، بهینه سازی آن برای پیاده سازی به صورت نهفته آغاز شد. با طراحی ساختمان داده مورد نیاز، کد زدن بخش مسیرداده صورت گرفت و به موازات آن کنترل کننده بر اساس فلوچارت طراحی شد. پس از شبیه‌سازی مراحل پیاده سازی بر روی برد هدف دنبال شد.



printed by MegaPrint Inc. www.postersession.com