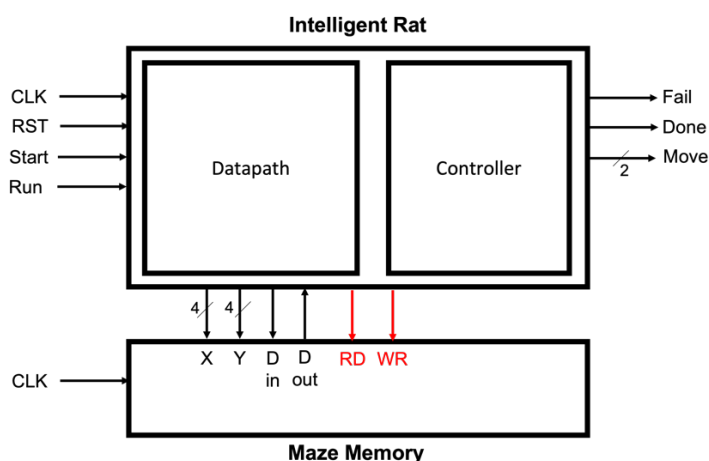




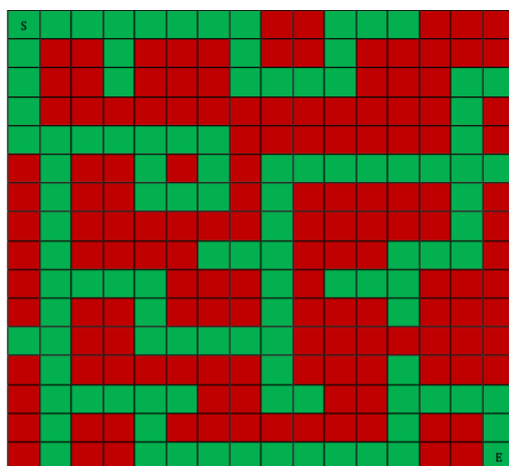
هدف از این تمرین کامپیوتری طراحی یک موش در هزارتو (Rat in Maze) است.

- مطابق شکل زیر، سیستم از دو ماجول حافظه‌ی هزارتو (Maze Memory) و موش هوشمند (Intelligent Rat) تشکیل شده است.



- ماجول حافظه‌ی هزارتو

- این ماجول یک حافظه‌ی 16×16 است که نقشه‌ی هزارتو را در خود ذخیره کرده است.
- وضعیت هر خانه از هزارتو با یک بیت نمایش داده می‌شود:
 - مقدار ۱ نشان‌دهنده‌ی دیوار است.
 - مقدار ۰ نشان‌دهنده‌ی فضای خالی است.
- نقطه‌ی شروع در گوشه‌ی پایین سمت چپ (مختصات ۰ و ۰) و نقطه‌ی پایان در گوشه‌ی بالا سمت راست (مختصات ۱۵ و ۱۵) قرار دارد.
- شکل زیر یک هزارتوی نمونه را نشان می‌دهد. رنگ سبز نشان‌دهنده‌ی فضای خالی (مقدار ۰) و رنگ قرمز نشان‌دهنده‌ی دیوار (مقدار ۱) است.



- در شروع کار نقشه‌ی هزارتو از یک فایل خوانده شده و در حافظه‌ی هزارتو ذخیره می‌شود.
- برای خواندن محتوای یک خانه از هزارتو (برای تشخیص دیوار یا فضای خالی) از دو ورودی چهار بیتی X (آدرس سطر) و Y (آدرس ستون) استفاده می‌کنیم.
- در صورت نیاز امکان نوشتن در یک خانه از فضای هزارتو هم وجود دارد. توجه داشته باشید که این امر برای لود کردن نقشه در حافظه‌ی هزارتو نیست و فقط در صورت نیاز برای اجرای الگوریتم یافتن مسیر می‌توانید از این قابلیت استفاده کنید.
- خواندن از حافظه‌ی هزارتو به صورت آسنکرون و نوشتن در آن به صورت سنکرون با کلاک انجام می‌شود.

• ماجول موش هوشمند

- هدف آن است که موش با انجام ترتیبی از حرکتهای رو به بالا، راست، چپ و پایین بتواند از هزارتو خارج شود.
- انتخاب حرکت توسط موش طبق اولویتهای زیر انجام می‌شود:
 - اولویت اول حرکت رو به بالا
 - اولویت دوم حرکت به سمت راست
 - اولویت سوم حرکت به سمت چپ
 - اولویت چهارم حرکت رو به پایین
- الگوریتم حل مسئله به صورت عقب-گرد (Back-Track) و به کمک یک پشته‌ی ۲ بیتی از حرکات انجام می‌شود
 - در ابتدای کار پشته تهی است و موش در موقعیت اولیه قرار دارد.
 - موش ابتدا جهت‌های مختلف را به ترتیب (بالا، راست، چپ و پایین) واری می‌کند. اولین جهتی که خالی باشد (یعنی فاقد دیوار باشد) انتخاب می‌شود و در پشته push می‌شود، سپس بیت مربوط به خانه‌ی فعلی موش یک می‌شود، یعنی آن خانه از آن پس مانند یک دیوار خواهد بود (برای جلوگیری از حرکت همیشگی در مسیرهای بسته).
 - اگر موش به نقطه‌ای برسد که تمام جهات اطراف آن دیوار باشد یا قبلاً امتحان و منجر به شکست شده باشد، باید عمل عقب‌گرد صورت گیرد.
 - در این صورت اگر پشته خالی باشد، مسئله فاقد جواب است و سیگنال خروجی به نام Fail فعال می‌شود. اگر پشته خالی نباشد یک عمل pop از روی آن صورت می‌گیرد و موش یک حرکت در خلاف جهت pop شده انجام می‌دهد، خانه‌ی مقصد را از حالت دیوار بودن خارج می‌کند و جهت بعدی مقدار بازگشتی پشته را امتحان می‌کند.
 - این رویه تا جایی ادامه پیدا می‌کند که موش به لبه‌ی هزارتو برسد و بتواند از آن خارج شود که در این صورت سیگنال موفقیت به نام Done فعال می‌شود. در این حالت مسیر طی شده (از آخر به اول) در استک قرار دارد.
 - پس از حل مسئله، مسیر طی شده در یک صف (FIFO) ذخیره می‌شود و در دفعات بعدی از این مسیر ذخیره شده برای حرکت استفاده می‌شود.
- ورودی‌های موش هوشمند:
 - سیگنال RST: تمام رجیسترهای داخل مسیرهاده را ریست می‌کند، استک و صف موجود در مسیرهاده را تخلیه می‌کند و موش را به نقطه‌ی ابتدایی برمی‌گرداند و آماده‌ی دریافت فرمان شروع می‌شود.
 - سیگنال Start: الگوریتم یافتن مسیر را آغاز می‌کند و مسیر پیدا شده را در یک صف ذخیره می‌کند.
 - سیگنال Run: موش را به نقطه‌ی ابتدایی برمی‌گرداند و حرکت را بر اساس مسیر ذخیره شده در صف نشان می‌دهد.

○ خروجی‌های موش هوشمند:

- سیگنال Fail: در صورتی که مسیری از نقطه‌ی شروع به نقطه‌ی پایان وجود نداشته باشد، این سیگنال فعال می‌شود.
- سیگنال Done: در صورت پیدا کردن یک مسیر از نقطه‌ی شروع به نقطه‌ی پایان، این سیگنال فعال می‌شود.
- سیگنال Move: در صورت فعال شدن سیگنال Run، حرکت‌های مسیر پیدا شده به صورت مرحله به مرحله بر روی این خروجی نشان داده می‌شود.

روش ارزیابی:

- پیاده‌سازی موش در هزارتو ۱۰۰ نمره دارد
 - ۲۵ نمره طراحی مسیر داده و واحد کنترل (به همراه کد ورپلاگ باید بارگذاری شود)
 - ۱۵ نمره روش کدینگ (مسیر داده به صورت ساختاری و واحد کنترل به روش هافمن)
 - ۴۰ نمره صحت طراحی با هزارتوی طراحی شده توسط شما
 - ۲۰ نمره صحت طراحی با هزارتوی طراحی شده توسط دستیاران آموزشی