

آزمایش امتیازی - آزمایشگاه معماری کامپیوتر

موضوع آزمایش: مرتب کننده ۸ عدد ۴ بیتی با استفاده از الگوریتم Bubble sort

تراشه‌های مورد استفاده:

۲۴ تراشه ۷۴۱۵۷، ۸ تراشه ۷۴۱۹۱، ۲ تراشه ۷۴۸۵، یک تراشه ۴۰۱۳، یک تراشه ۷۴۱۳۸، ۸ تراشه ۷۴۱۵۱، تعدادی تراشه AND, OR, NOR, NAND، ۱۰ عدد BCD seven segment

شرح آزمایش:

در این آزمایش قصد داریم که با استفاده از الگوریتم Bubble sort، یک مرتب کننده با قابلیت مرتب کردن صعودی ۸ عدد ۴ بیتی بسازیم. بدین منظور برای ورودی دادن به مدار، از ۳۲ LogicState در سمت چپ مدار استفاده شده و دو ورودی Start و Reset نیز داریم. با نگه دادن ورودی و نگه داشتن Start مدار شروع به کار می‌کند و مراحل انجام شدن Bubble sort در Seven segment ها قابل رویت است. پس از اتمام سورت، سیگنال خروجی End برابر ۱ می‌شود.

نحوه پیاده‌سازی، همانطور که می‌دانیم در Bubble sort، ابتدا بزرگترین عدد به آخر دنباله می‌رود، بعد یکی مانده به بزرگترین عدد و الی آخر. بدین منظور برای نمایش وضعیت مراحل الگوریتم، از دو BCD seven segment برای هر یک از متغیرهای دو حلقه راجع در الگوریتم Bubble sort استفاده شده است که می‌توان با آن فهمید که الگوریتم در کدام مرحله است. در هر گام i ام از مرحله j ام، عدد i ام با عدد i+1 ام مقایسه می‌شود و در صورتی که عدد i ام بزرگتر باشد، این دو عدد swap می‌شوند. سپس i++ می‌شود و اگر j == i شد، j-- شده و i = 0 می‌شود.

بنابراین در ورودی ۸ رجیستر مربوط به اعداد ما، یک MUX سفارشی شده قرار دارد که ۴ مدل ورودی می‌تواند به رجیسترهای ما بدهد، یکی ورودی موازی است که در ابتدا به مدار می‌دهیم، یکی مقدار فعلی خود رجیستر، یکی مقدار رجیستر قبلی و دیگری مقدار رجیستر بعدی.

برای اینکه بدانیم که در این گام، باید کدام اعداد را با هم مقایسه کنیم، ۸ عدد MUX به ۱ و ۳ ورودی سلکت داریم که مشخص می‌کنند که کدام دو عدد با هم مقایسه شوند. یک دیکودر نیز داریم که تعیین می‌کند در صورت کوچکتر بودن عدد بعدی، سیگنال write کدام رجیستر(ها) فعال شود. دو شمارنده نیز داریم که در بالای مدار قرار دارند و نقش متغیرهای i و j را در کد معمول الگوریتم Bubble sort بازی می‌کنند. شکل کلی مدار و همچنین شکل MUX سفارشی شده در زیر قرار گرفته است:

