

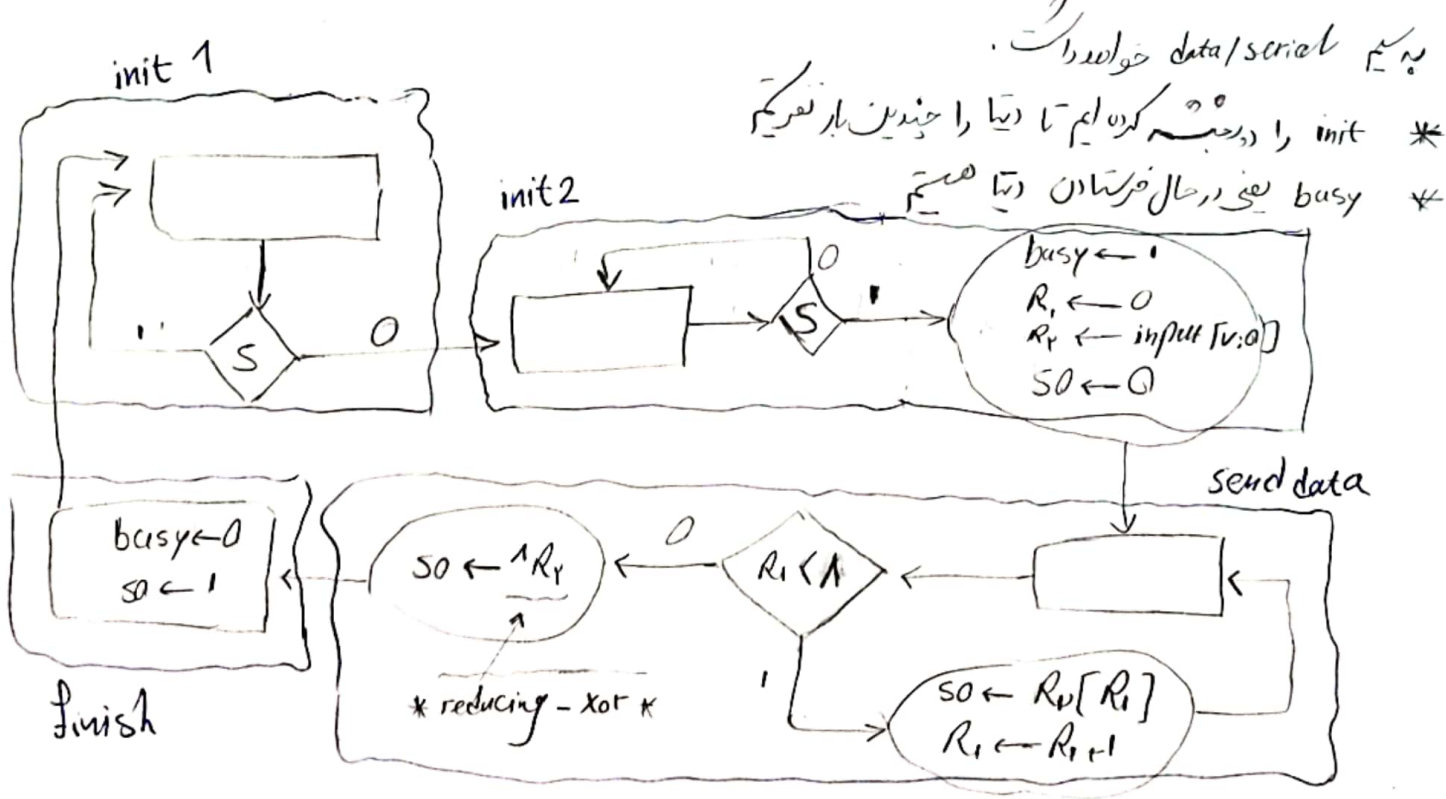
این سخت افزار نیاز به دو ماژول دارد: یک ماژول گیرنده و یک ماژول فرستنده که این دو ماژول توسط یک بایوس ارتباط برقرار می کنند. (۱) سیم data/serial که در هر سابلینک فلاک، ماژول فرستنده و بایوس فرستنده و ماژول گیرنده یک بیت رای گیری (۲) سیم ground که سطح ولتاژ این دو را یکسان می کند. (۳) CLK به طور کلی بیت کنترلی S ایندونه کاری کند که زمانی که برابر ۱ است، روی standby است و کاری انجام نمی دهد اما وقتی که بیت S به صفر تغییر کرد، بایوس، ارتباط می زنم و به ترتیب دیتاهای در براری فرستیم:

① data[0] ② data[1] — data[v] ④ parity و سپس این یک را ① می گیریم

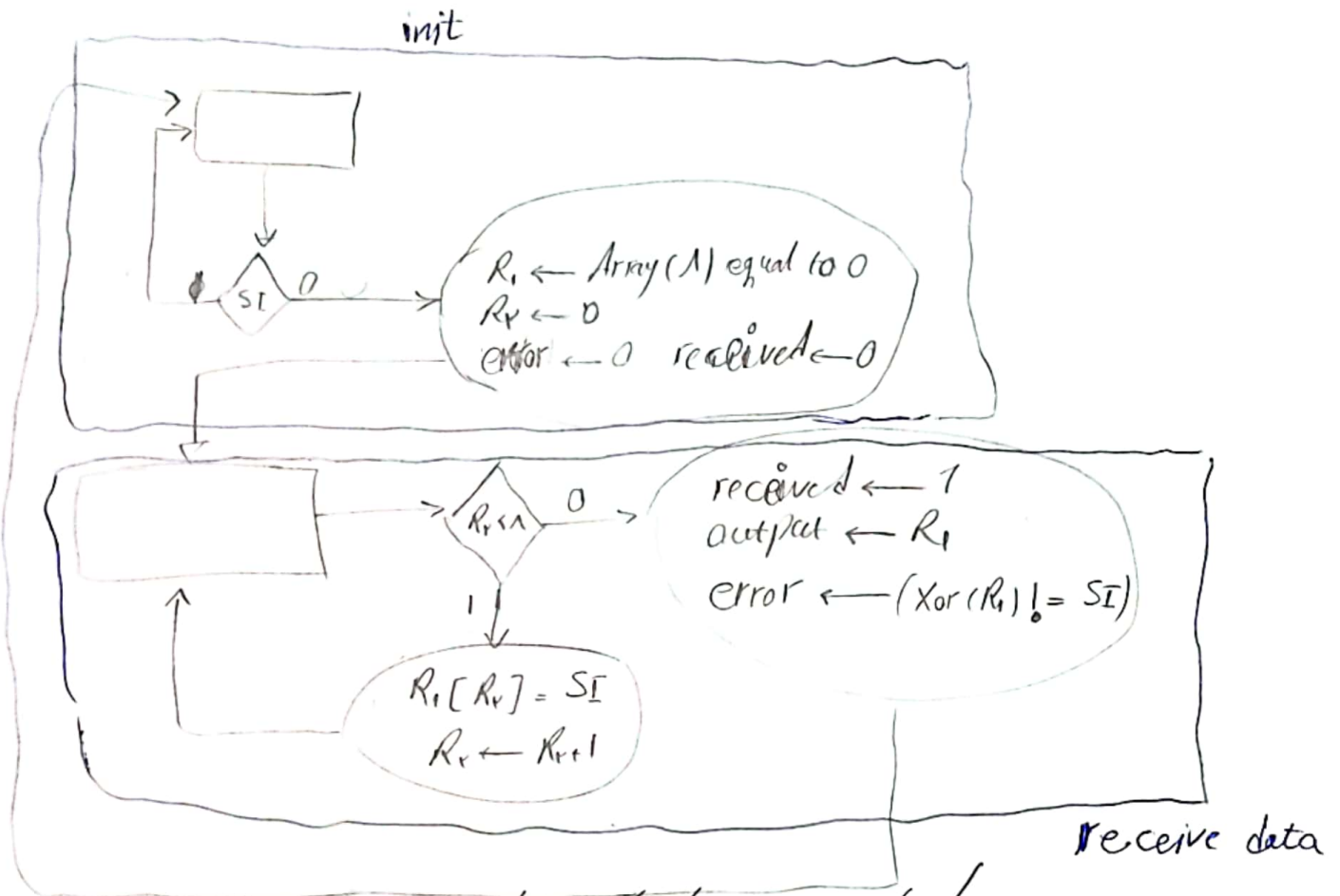
متوقف شود و منتظر درباره صفر شدن بماند. «SO هم همینطور کاری کند»

استاندارد بیت parity برای پیدا کردن برخی از ایتاهای اشتباه است که به دلیل وجود noise پدید می آید. در صورت read اشتباه سیم error گیرنده را فعال می کنیم.

فرستنده، input[v:0]، CLK، S را به عنوان ورودی می خواند و در SE و همچنین SE را به عنوان خروجی



گیرنده، SI، CLK و به عنوان ورودی خواسته شد و output (v:0)، error و received را به عنوان خروجی خواسته شد.



* در این قسمت آخری توانیم یک بلاک finish اضافه کنیم که این بلاک به معنی اتمام عملیات و صرفاً حلی ما را می‌گیرد که یک فیدبک روی init حاسب می‌کند!

* بیت ۸ به پای $X < 8$ می‌توانیم از $X = 8$ استفاده کنیم زیرا سخت‌افزار کمپلتری دارد

* بیت ۹: تمام کارهایی که در یک بلاک انجام داریم، قابل اتمام در یک CLK هستند.
برای روشی indexing ای که استفاده می‌کنیم می‌توان استدلال کرد که ۸ بیت را کنار هم داریم باید mem بتی که می‌خواهیم را انتخاب می‌کنیم.

* سیم‌های cycle accurate به سبک تر اند!