

«پنجم خدا»

«9731098» «رایان قرنجی»

«فرآیند ورودی‌ها و خروجی‌ها و خودی عملکرد خواسته شده»

① ماژول Multiplicand: یک رجیستر هستی که از DFF ساخته شده و به آن هستی

DFF فراخوانی شده و DFF ها از Reset فعال باشد، خروجی شون صفر هست و اگر نباشه، خروجی

D هست که یک سینال Load هم داره که از واحد کنترل می‌گیرد و بلافاصله می‌گیره چون فلیپ‌فلایپ‌های

درونی با Clock کار می‌کنند و Reset هم برای اینکه DFF ها را منو کنند و Load که A برود به رجیستر

اگر فعال باشد A نباشه، فعلی حالت قبل است.

② ماژول Adder: که از هستی Full Adder ساخته می‌شود و هستی بیت است که RA, RB

را می‌گیرد و Add\_Out و Carry هم می‌شود Out که از درتا Xor، درتا And و یک or

تشکیل شده است.

③ ماژول Controller: که محاسبات ورودی‌ها و بیت‌های خروجی‌ها را می‌گیرد و چون دستور 7 می‌فرستد و

ورودی‌هایی که می‌گیرد که بهترین بیت ماژول Multiplier\_Result، reset، clk و Start را ورودی

می‌گیرد و خروجی‌ها Stop و Shift و App به ماژول Multiplier\_Result است و دستوراتی که Load

که در ماژول‌های Multiplicand و Multiplier\_Result و Load می‌شود: ماژول M-Result، 77 بیت است.

عدد B و A

رجیستر است که هم می‌تواند خودش را به سمت راست تغییر دهد و هم مقدارش را update کند و با رجیستر

ماژول Multiplier جمع می‌شود. ماژول Controller با استفاده از ماشین حالت، حالت‌های مختلف

شروع به کار کردن، حالت اولتیمیت، بیت Add & Shift که این حالت‌ها را تعریف کردیم و بعد به تغییر

Counter & Counter به وضعیت برسد، انجام بدهد) و یک State (که بدانیم به کدام حالت هستیم).

یک NextState (که حالت بعدی را نشان دهد)، ماشین حالت شامل دو بخش ترکیبی هستند که

بیت ترکیبی می‌گردد. ~~Reset~~ بودیم، ~~به~~ IDLE، باشد، نبودیم به حالت بعدی برویم ازای هر کلاک.

اینکه هر State می‌است در بخش ترکیبی مدار ساخته می‌شود. اولاً دستوراتی که در خروجی بودند را معرفی کنیم و بر اساس

اینکه State می‌تواند، صادر شوند می‌کنیم، ~~که اگر در IDLE بود، Stop را یک کرد~~ و Counter هم می‌گردد.

سیگنال Start آمد، ضرب باید شروع بشود که اگر نیامده، در همان IDLE بماند. اگر نه Initial بودیم، سیگنال Load

را فعال کند که در دست بودیم، بسته به کوچکترین بیت M\_Reset که اگر می‌بود به سبقت برود و اگر ~~به~~ Add نرود

و سپس به وضعیت ADD بردیم، و فعال add فعال بشود و به وضعیت برگردیم، فعال  
وضعیت فعال بشود تا اینکه Counter کی اتمام بشود، اگر Counter به وضعیت برگردیم، به Start اول و Stop ولی  
اگر فرسیده بود برگردیم به حالت Test.

(4) ما عمل Multiplier\_Reset : PFF دارد و باید مقدار ندارد و توان Shift و Add کردن هم داشته باشد،

و وقتی که Load فعال بود 4 عدد B به درجست بیت پس و وقتی Shift فعال بود، 7 بیت را به راست

Shift بدیم و اگر Add فعال بود، باید وضعیت بیت بالا را مساوی مقدار می آید قرار دهیم.



ولی اگر Add فعال نبود، هست بیت بالا را می‌شد و در خودش بیخذه است. کو چنانچه بیت 7 و LSB است که باید

به Control و RC هم حاصل ضرب است و RB هم هست بیت بالا است. ورودی ای که از بیت جمع نشت می‌آید،  
(7 بیت پایین)

Sum است و سیگنال‌های Add، Reset، Shift، Load، B و Sum ورودی چون هستند و RC،

RB و LSB هم خروجی‌ها. متغیر Temp، خروجی DFF ها است و 17 بار از DFF ها استفاده  
(از موقت تا آخر)

شده است. اگر سیگنال Shift فعال بود، مقدار DFF همان Temp قبل است به شرط آنکه یکی به

راست Shift داده شود. اگر ADD فعال بود، ورودی DFF می‌شود آن هست بیت که از جمع نشت می‌آید که هست  
(هست بیت بالا)

بیت پایین ثابت است و اگر Load فعال بود، 8 بیت بالا ثابت و همین Temp باشد و هست بیت پایین را مساوی

B کرد. در غیر اینصورت اگر هیچ‌کدام از Shift، Add و Load فعال نبودند، ورودی DFF ها همچون Temp باشد.

است  
⑤ مازول حل: ورودی‌ها CLK، Reset، Start هستند و اعداد A، B که ضرب شده و RC خروجی آن

و Stop که ضرب را تمام کند. سیم‌های داخلی ارتباطی هم با Wire تعریف شده‌اند. یکی Add و Load و LSB

shift, Cout یک بیت و یکی RA, RB, Add, Out یک بیت است. و بعد چهارم مازول قبلی به اسم‌های M\_Cand

مکانیسم

Subject .....

Date .....

6. Adder و M\_Res فراخوانی شدند و سیستم‌های واسطه را به‌شون وصل کردیم.

Test Bench: به ازای ورودی‌ها reg و به ازای خروجی‌ها wire تعریف می‌کنیم و مازول‌کل را فراخوانی کرده و

مقداردهی به سیگنال‌های ورودی (برای 20ns، دستورات بالا، CLK هر ثانیه‌ای عوض شده و not می‌شود و بعد از

20ns دستورات پائین اجرا می‌شود.