

۱- منظور از Set on less than در واحد alu این است که در این خط کنترلی، داده ای که کوچکتر است روی خروجی قرار بگیرد.

۲- گاهی اوقات برای برخی مدارهای pipeline لازم می شود مدارهای ترکیبی نیز خروجی خود را لچ کنند در اینجا همانطور که گفته شده ، ماژول alu شما در لبه ی بالارونده کلاک خروجی را لچ می کند.

۳- برای ایجاد حافظه ها از رجیستر دو بعدی استفاده می شود به صورت [16:0]memory[31:0] که عددی که جلوی حافظه نوشته شده عمق حافظه و عدد دیگر تعداد بیت عدد هایی که داخل حافظه قرار دارند می باشد.

۴- در فاز دوم پروژه برای قسمت اول و قسمت دوم ابتدا از کامل انجام شدن دستورالعمل قبلی اطمینان پیدا کنید سپس instruction دستورالعمل بعدی را load کنید (برای اینکار می توانید یک شمارنده بگذارید که پس از گذشتن تعداد کلاک مشخص که دستورالعمل قبلی کامل شد ، واحد instruction کلاک بخورد و instruction دستورالعمل بعدی load شود)

۵- در فاز دوم برای قسمت اول همانطور که گفته شده، لازم است در واحد رجیستر بانک مقدار رجیستر صفر را مقدار ثابت صفر قرار دهید ($0 \leq \text{databank}[0]$) تا در اجرای دستورالعمل ها مشکلی ایجاد نشود. همچنین در قسمت دوم این فاز قبل از اینکه دستورالعمل load را انجام دهید در خانه ی ۱۳ data memory مقدار ۳ را بنویسید تا با این دستورالعمل این مقدار را load کنید.

$$1. \text{lw } R_1, 10(R_2) \quad \Rightarrow \quad R_1 = \text{Mem}(R_2 + 10) \quad \text{I-Type}$$

$$2. \text{beq } R_2, R_1, 20 \quad \Rightarrow \quad \text{if } R_2 = R_1 \text{ then } PC = PC_{\text{previous}} + 20 \quad \text{I-Type}$$

۶- معماری MIPS که پیاده سازی کردید دارای ورودی clk و خروجی این معماری را خروجی واحد هایی که در گزارش پروژه از شما خواسته شده تا مقدارشان دیده شود قرار دهید. و testbench ها را با توجه به این ساختار ها بنویسید. (یا به هر شکلی که بتوانید اسکرین شات هایی که در گزارش پروژه گفته شده را ببینید و از درستی mips خود مطمئن شوید)