

دانشکده مهندسی کامپیوتر

دانشگاه اصفهان

**گزارش تمرین سوم درس طراحی کامپیوتری سیستم‌های دیجیتال**

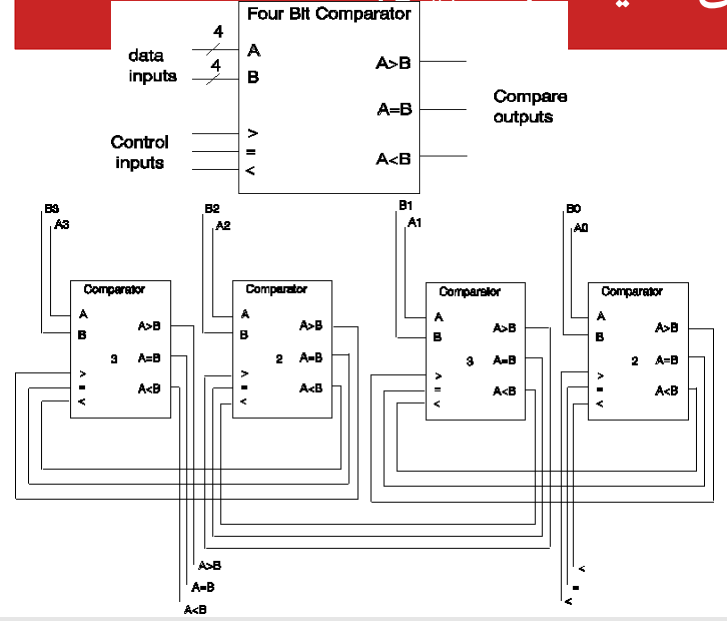
**شرح پیاده‌سازی**

**امیرارشیا همت**

**۹۸۳۶۱۳۰۶۳**

**توضیح پیاده‌سازی:**

با توجه به شکل یک مقایسه‌گر چهاربیتی که در پایین آمده‌است، پیاده‌سازی انجام شده‌ است:



براین اساس می‌توان دریافت باید چهار مقایسه‌گر تک‌بیتی را بصورت سری کنار هم قرار دهیم و در نهایت خروجی مقایسه‌گر چهارم، همان پاسخ ما می‌باشد.

در ابتدا باید مقایسه‌گر تک بیتی را پیاده‌سازی نماییم. با توجه به اسلاید‌های استاد، پیاده‌سازی یک مقایسه‌گر تک بیتی انجام شده است و در کد به نام one\_bit\_comparator آورده‌ شده است. برهمین اساس با توجه به مقایسه‌گرهای تک‌بیتی باید یک مقایسه‌گر چهاربیتی ساخته و چهار مقایسه‌گر تک بیتی را در آن قرار دهیم. به همین منظور ابتدا یک entity به‌نام four\_bit\_comparator ساخته و برای ورودی دو bit\_vector چهار بیتی به نام‌های input1 و input2 تعریف می‌کنیم.

\*دلیل عدم استفاده از std\_logic\_vector این است که با توجه به اینکه مقایسه‌گر تک‌بیتی ما از نوع bit است و نه std\_logic درصورت استفاده از std\_logic\_vector به مشکل برخورده و باید حتما از bit\_vector استفاده کنیم.

حال نکته‌ مهم بعدی ورودی‌های entity مقایسه‌گر چهاربیتی‌مان هست. ما سه‌ پایه gt, eq, lt را برای ورودی به همراه دو ورودی چهاربیتی که قرار است با هم مقایسه شوند به عنوان ورودی درنظر گرفته‌ایم(هرچند که می‌توان از سه پایه gt, eq, lt صرف نظر کرد اما با توجه به اسلاید‌های استاد من به عنوان ورودی قرار داده‌ام)

**Four-bit comparator**

a\_gt\_b

a\_eq\_b

a\_lt\_b

Input1 4bit

Input2 4bit

gt  
eq  
lt

سپس چهار کامپوننت از مقایسه‌گر تک‌بیتی ساخته و با سیم‌هایی که تعریف کردیم خروجی هرکامپوننت را به ورودی کامپوننت بعدی وصل می‌کنیم(دو ورودی اول کامپوننت هم که همان بیت‌هایi ام دو bit\_vector ما هستند) و خروجی کامپوننت آخر را هم برابر خروجی entity مقایسه‌گر چهاربیتی قرار می‌دهیم.

اتصال کامپوننت‌ها با هم به کمک سیم‌هایی است که تعریف شده‌اند و درواقع به کمک این‌سیم‌هاست که خروجی مقایسه‌گر قبلی در ورودی مقایسه‌گر بعدی وارد می‌گردد.