La stringa negli shift register viene shiftata verso dx. Lo stato di shift è messo dopo il primo confronto perché di default come uscita degli shift registers già ci sono i bit più significativi delle due stringhe. Per come è costruito, lo shift register, quando è attivo il segnale di load, inverte la stringa in ingresso. Una implementazione analoga sarebbe potuta essere quella di implementare uno shift verso sx, per far sì che sia sempre prima il bit più significativo ad essere shiftato.

Notiamo il funzionamento dell'handshaking: B legge dalla memoria e invia un segnale di richiesta ad A. Dopo un colpo di clock, A legge il segnale di req e va nello stato di send ack x, dove invia l'ack che testimonia la corretta ricezione della prima stringa. Successivamente, viene abilitato il segnale di load per lo shift register, e A va in uno stato di attesa, in attesa che lo shift register venga caricato. Lo shift register viene caricato sul fronte di discesa del colpo di clock successivo a quello in cui ho alzato il load. La stessa cosa viene ripetuta per Y, dove si attiva lo specifico segnale per il load dello shift register y. Il segnale shift è unico per entrambi gli shift register in quanto lavoro sempre su bit omologhi.

Quando poi A effettua la comparazione delle due stringhe, si nota come nel frattempo B invia già la prossima stringa in ingresso allo shift register, ma questa stringa non verrà letta fin quando non termina la comparazione. Quindi A rimane in attesa dell'ack di B, che nel frattempo sta confrontando i bit.

Per unire la funzionalità di stabilire se una stringa e più grande rispetto all’altra e contare il numero di bit omologhi, si sono introdotte due abilitazioni differenti per il comparatore di bit, una in cui valuta >,< e = e una in cui valuta solo =. Una volta che si alza il flag > o <, significa che una delle due stringhe è Maggiore dell’altra e quindi spengo la modalità di funzionamento < e > e procede solo con quella =, per continuare a contare il numero di bit omologhi uguali. Successivamente, nello stato end check dell’automa del comparatore, una volta che ho finite tutte le 8 cifre, verifico se il numero di bit omologhi è < 4 e in quel caso alzo l’abilitazione di un contatore interno.