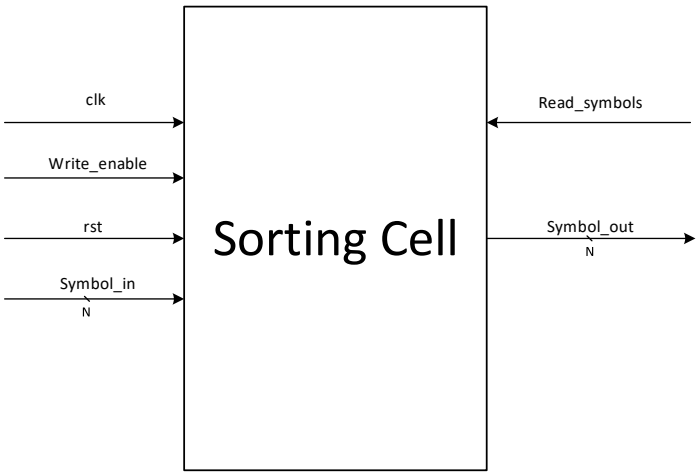
Sorting algorithm hardware accelerator

Gli algoritmi di ordinamento (sorting) sono stati largamente studiati negli ultimi anni. A diversi algoritmi, eseguiti da microcontrollori, corrisponde in generale un diverso grado di complessità o ordine dell'algoritmo. Tipicamente andando ad eseguire un algoritmo su un singolo core è impossibile ottenere un algoritmo di ordine lineare **O(N)**. Come succede spesso per vari tipi di applicazioni, alcuni compiti particolarmente onerosi da eseguire per un microcontrollore possono venire realizzati in logica programmabile, in modo da ottimizzarne le performance. Si richiede di progettare un circuito digitale col compito di implementare un semplice algoritmo di ordinamento. L'interfaccia del circuito da progettare è la seguente:



Sia M il numero massimo di elementi (simboli) ordinabili ed N la dimensione dei singoli elementi. M e N sono dei generici disponibili per la configurazione da parte dell'utente. La sorting cell al suo interno ha il compito di tenere memoria degli ultimi M elementi ricevuti in ingresso e ordinarli al suo interno secondo ordine crescente considerando gli ingressi, di tipo std_logic_vector, come unsigned. I simboli in ingresso vengono letti e processati dall'algoritmo di ordinamento solo se il segnale di write enable è alto. La procedura di lettura in uscita dei simboli ordinati viene attivata quando read_symbol viene asserito per un ciclo di clock, e consiste nel leggere tutti gli elementi, presenti e ordinati all'interno del blocco, in ordine crescente.

È richiesto di trattare le varie possibili situazioni di errore come meglio si crede, documentando le scelte fatte. In particolare è necessario pensare a:

- Come gestire un conflitto di contemporanea lettura e scrittura
- Come gestire eventuale overflow del blocco
- Come gestire eventuale richiesta di lettura senza che non ci sia nessun elemento interno al blocco.

La relazione finale del progetto deve contenere:

- Introduzione (descrizione encoder, possibili applicazioni, possibili architetture, etc.)
- Descrizione dell'architettura selezionata per la realizzazione (diagramma a blocchi, ingressi/uscite, etc.)
- Codice VHDL (con commenti dettagliati)
- Test-plan e relativi Testbench per la verifica
- Interpretazione dei risultati ottenuti nella sintesi automatica su piattaforma Xilinx FPGA Zync in termini di massima frequenza di clock (cammino critico) e elementi utilizzati (slice, LUT, etc.) al variare di M e N. Commentare eventuali messaggi di warnings.
- Conclusioni