Quest’implemetazione dello Z80 su FPGA presenta dei problemi, in alcuni casi facilmente risolvibili, di cui bisogna tenere conto durante l’utilizzo.

Per prima cosa bisogna far notare il rapporto tra le frequenze di CLK\_FPGA e CLK. I due in questo caso sono in rapporto minimo pari a 20:1. Durante l’uso bisogna tenere conto che l’approssimazione dei latch con FFs temporizzati su CLK\_FPGA vale fintantochè il rapporto tra le frequenze rimane elevato. Per cui se si volesse far andare Z80X fino ai 20MHz che sono garantiti dal datasheet delle ultime versioni CMOS [Z80.pdf p.1] bisogna fornire un CLK\_FPGA con frequenza almeno dieci volte maggiore.

L’istruzione OUTI è difettiva. L’istruzione esegue la copia di una locazione di memoria su una periferica incrementando l’indirizzo. L’istruzione nel decoder è eseguita dallo stesso blocco che esegue tutte le operazioni di copia di blocchi di memoria anche ripetuti cioè OUTI, OTIR, OUTD, OTDR, LDI, LDIR, LDD e LDDR. In questo insieme OUTI è l’unica che richiede 17 T-cycles contro i 16 delle controparti[Z80.pdf p.18]. Per cui nello Z80X, l’istruzione viene eseguita come le altre con 16 T-cycles considerando la presenza del ciclo in più un errore.

Siccome la decodifica delle istruzioni è stata basata sul buon senso assieme alle informazioni presenti sul datasheet con l’intuizione della divisione in cicli simili, non è rispettat la reale divisione e durata dei singoli M-cycles. Per cui è sempre garantita la durata totale in T-cycles come da datasheet e spesso anche il numero di M-cycles corrisponde per difetto ma non rispetta la vera divisione dei T-cycles per ogni M-cycles. Significa che lo Z80X messo a confronto con uno Z80 originale, avrà la stessa durata delle istruzioni ma potrebbe non accogliere nello stesso momento le richieste del bus poiché i cicli macchina dello Z80X potrebbero iniziare e finire prima o dopo rispetto a quelli dello Z80.

In ultima analisi vi è l’occupazione di SLICEs dell’entity.  
La decisione di creare un’ALU con tre diverse sezioni per ogni operazione richiede uno spazio che può essere ridotto usandone solamente uno assieme a più logica di controllo.  
La struttura del DECODER occupa molto spazio a causa del controllo di tutte le condizioni che non avviene per mezzo di memorie ma di singole LUT all’interno delle SLICE. Una futura implementazione migliore può essere fatta usando una BRAM come se fosse un PLA e indirizzarla attraverso la parola generata dalla giustapposizione di tutti gli ingressi al DECODER. Questa soluzione ridurrebbe indubbiamente il numero di SLICEs usate però potrebbe incorrere nel problema che le BRAM a disposizione possano non bastare per tenere tutte le parole richieste. Inoltre richiederebbe uno sforzo maggiore nella programmazione poiché richiederebbe di lavorare con anonime parole di lunghezza considerevole ad anonimi indirizzi di memoria aumentando la difficoltà di debug.