L’ALU e due INC/DEC sono gli elementi principali di calcolo dello Z80X. Nell’architettura dello Z80 compare solo l’ALU mentre dalle informazioni sul reverse engineering si vede che l’ALU in realtà è a 4 bit e vi è un INC/DEC a 16 bit. Per semplicità di progetto, per le operazioni di incremento e decremento a 8 bit ho aggiunto un INC/DEC dedicato.

ALU

Figura X – Schema dell’ALU.

L’ALU presenta dei registri interni che campionano il valore dell’accumulatore, del secondo operando e del registro di stato F. Questi registri vengono aggiornati quando CLK\_NEDGE è attivo e sull’uscita il risultato è sempre presente anche se solitamente viene campionato con CLK\_PEDGE. Questo permette di usare i bus ad altri scopi. I latch dell’ALU, per necessità di temporizzazione con altri eventi, sono aggiornati anche da STATE\_CNG che è un segnale generato dalla sezione di controllo che si attiva per un ciclo di clock se lo stato della FSM Main cambia.

L’implementazione dell’ALU considera sia le informazioni sull’organizzazione reale dello Z80 sia la semplicità di implementazione. Di conseguenza per le operazioni aritmetico-logiche, vi sono tre unità che lavorano in contemporanea. Una che calcola il risultato su 9 bit invece che 8 bit estendendo il segno, il risultato serve sia per generare il flag di overflow sia come risultato effettivo. Un’altra calcola la stessa cosa ma senza estendere il segno per cui il nono bit rappresenta il flag di carry out. Infine una a 5 bit che considera la parte bassa a 4 bit dei due operandi senza estensione del segno generando così il flag dell’half carry.

L’ALU svolge 27 differenti operazioni selezionabili per mezzo del vettore OPT, abbrev. di Option, attraverso codici riconducibili all’istruzione che effettua la chiamata. Nel caso in cui l’operazione selezionata non sia riconosciuta, l’ALU esegue un NOP.

Il primo gruppo contiene le 7 operazioni aritmetico-logiche a 8 bit e CP, abbrev. di Compare. Quest’ultimo viene implementato come una differenza che modifica i flag ma restituisce in uscita il valore di A non modificato.  
Le istruzioni vengono selezionate per mezzo della giustapposizione del prefisso 000- con il campo corrispondente, bit da 5 a 3, dell’istruzione.  
Per la loro controparte a 16 bit, il DECODER prima fa eseguire l’operazione a 8 bit come richiesto e poi esegue l’operazione sugli altri 8 bit sempre con l’uso del carry in.

Il secondo gruppo contiene le 4 operazioni di rotazione dell’accumulatore. Queste operazioni vengono svolte per semplice riassortimento dei bit e calcola i bit del flag come se fosse una normale operazione aritmetico-logica. Il valore di OPT si genera con il prefisso 001- seguito dal campo corrispondente, bit da 5 a 3, dell’IR. Ad appendice a questo gruppo vi sono le 2 istruzioni di rotazione per cifre BCD, le istruzioni RRD e RLD.

Il terzo gruppo contiene 3 istruzioni di gestione dell’accumulatore: DAA, CPL e NEG.  
DAA è l’abbrev. di Decimal Adjust Accumulator e serve a riportare in formato BCD l’accumulatore, se è stato generato per mezzo un’operazione aritmetica da due numeri BCD. Fa uso dei flag H, C ed N e dal reverse engineer si è visto che principalmente segue questa logica:  
se N è 1 tutte le operazioni successive sono di sottrazione invece che di somma;  
se la parte alta del risultato nell’accumulatore è maggiore di 9 o C è settato, viene sommato 60H;  
se la parte bassa del risultato nell’accumulatore è maggiore di 9 o H è settato, viene sommato 6H;  
nei casi previsti si eseguono entrambe le somme.  
CPL è l’abbrev. di Complement ed esegue il complemento a 1 cioè la negazione logica dell’accumulatore mentre NEG, che è l’abbrev. di Negation, esegue il complemento a 2 cioè il cambiamento del segno.

Il quarto gruppo è formato dalle 3 istruzioni che effettuano operazioni sui bit: BIT, RES e SET.  
BIT effettua solamente il test del bit selezionato per cui esegue un’AND tra il negato del valore da testare e una maschera di tutti 0 tranne un 1 nella posizione del bit corrispondente. Dopodiché il risultato non viene dato in uscita ma vengono modificati solo i flag in particolare il flag Z. Mentre RES, abbrev. di Reset, effettua un’AND tra il valore e una maschera con di tutti 1 tranne uno 0 nella posizione del bit corrispondente e SET effettua una OR tra il valore e una maschera con di tutti 0 tranne un 1 nella posizione del bit corrispondente.  
Il codice per la selezione si ottiene con la giustapposizione del prefisso 0111- seguito dal campo corrispondente, bit 7 e 6, della parte basse dell’IR.

L’ultimo gruppo contiene le 8 istruzioni che effettuano le operazioni di scorrimento e rotazione su registri diversi dall’accumulatore. Oltre alle 7 documentate ho aggiunta anche SLL che si inserisce perfettamente nei codici di selezione. Questi vengono creati dalla giustapposizione del prefisso 100- con il campo corrispondente, bit da 5 a 3, dell’IR.

INCDEC e INCDEC16

Figura X – Schema dell’INCDEC e INCDEC16.

L’implementazione di INCDEC e INCDEC16 è la stessa a meno della differenza di lunghezza della parola. La selezione tra incremento e decremento avviene con la linea DECnINC che può essere collegata al bit 0 dell’IR che è 0 nel caso di incremento o è 1 altrimenti.

Le due entity presentano la stessa struttura a registri dell’ALU, per cui il valore del dato in ingresso viene campionato ad ogni attivazione di CLK\_NEDGE e STATE\_CNG ed il risultato viene solitamente letto in CLK\_PEDGE.

INCDEC presenta anche un ingresso per il carry in poiché genera anche il proprio valore del registro di stato F e necessità del valore precedente di carry in.