Come per la generazione del clock, il controllore delega la generazione, la memorizzazione e la traduzione degli snap a due entity Z80\_SNAPPER e Z80\_REPORTER.

Figura X – Schema dell’entity Z80\_SNAPPER.

L’entity Z80\_SNAPPER è basata su una FSM. La macchina rimane in un stato di attesa, WT, fintatochè il controllore non attiva il segnale di enable, EN, e la SNAP\_FIFO permette la scrittura, quindi FULL è disattivato.

In questo caso lancia un impulso di attivazione al CLK\_GEN ed entra in uno stato di attesa, IDLE. Esce da questo stato quando il segnale interno CNG si attiva. Questo segnale è generato dalla variazione di un vettore particolare creato dalla giustapposizione di sette campi con delle maschere generate dai corrispondenti sette bit dell’ingresso CNGENS. I sette campi sono nRESET, System Control, CPU Control meno nRESET, CPU Bus Control, A, DIN e DOUT. In questo modo si può avere uno snap solo al variare di segnali precisi come solo quelli del gruppo System Control per ridurre il numero di snap.

Quando CNG si attiva, la macchina attiva il segnale WR\_EN verso SNAP\_FIFO scrivendo lo snap compattato in 64 bit. Se la FIFO dopo quest’operazione attiva FULL, lo snapper attiva STOP bloccando CLKOUT ed entra nello stato di attesa, IDLE.

Figura X^2 – Schema dell’entity Z80\_REPORTER

Lo scopo dell’entity Z80\_REPORTER è solo quello di tradurre già in caratteri gli snap che sono presenti sulla SNAP\_FIFO. Per cui appena SNAP\_FIFO ha uno snap disponibile lo legge e genera il messaggio da stampare. Poi, fingendosi una ROM, lo fornisce a CHAR\_WRITER\_FIFO che lo carica carattere per carattere in REP\_FIFO. Appena finisce ricomincia da capo resettando l’entity.