

Designspecifikation: FPGA Z80

26 februari 2018

Noah Hellman: noahe116
Dennis Derecichei: dende301
Yousef Hashem: youha847
Jakob Arvidsson: jakar180

Innehåll

1	Analys	2
1.1	CPU	2
1.2	TI ASIC	2
1.3	IO	2
1.4	GPU	3
1.5	Programmering	3
2	Milstolpe	3
3	Blockschema	4
3.1	TI84	4
3.2	Z80	5

1 Analys

1.1 CPU

Vi ska bygga en halvt pipelinad z80 där hämt- och exekveringsfaserna överlappar. Programminnet och dataminnet är delat och ligger utanför processorn. Det finns en databuss som är 8 bitar och en addressbuss som är 16 bitar. Det finns även en kontrollbuss på 16 bitar som innehåller styrsignaler utifrån såsom INT och MEMRQ. Instruktionerna är mellan 8 och 32 bitar långa. Det finns olika adresseringsmoder såsom

- `ld a,55` - omedelbar, a laddas med 55,
- `jr 5` - relativ, PC adderas med 5,
- `jp 4578` - direkt, PC sätts till 4578,
- `ld a,(b)` - indirekt, a laddas värdet som b pekar på,
- `ld a,(IX+5)` - indexerad, a laddas med adressen IX pekar på plus 5.

Vi vill kunna ladda program via USB vid start av processorn. CPU:n ska kunna köra generella Z80-program och utföra alla uppgifter förutom det som IO-kontrollerna hanterar.

1.2 TI ASIC

En speciell krets kommer behövas för att koppla samman processorn med IO-enheterna som en TI84 förväntas ha. Z80:n använder sig av ett portsystem som denna kretsen ska implementera. Varje port är kopplad till en viss enhet och har ett visst syfte. Den här kretsen kommer dessutom agera som kontroller för tangentbord och VGA-skärm.

1.3 IO

IO kopplas till processorn med addressbussen, databussen samt styrsignaler. CPU:n meddelar att den vill komma åt IO via IORQ-signalen och lägger adressen till porten på addressbussen. En mux kopplar därefter databussen till rätt IO-enhet.

IO-enheterna kan göra interrupts med INT- och NMI-signalerna. Beroende på vilket mode som är inställt interagerar IO-enheten på olika sätt. Med

till exempel mode 0 kan IO-enheten efter INT-signalen placera en instruktion på databussen som CPU:n kommer exekvera nästa klockpuls. I mode 2 kan IO-enheten lägga en index på databussen till en avbrottsrutin som är lagrad i en tabell i minnet. Processorn kommer då hoppa till den rutinen efter den nuvarande instruktionen. Den moden som ska användas kan väljas med IM (interrupt mode) instruktionen. Dessa avbrott kan också stängas av och sättas på med DI och EI instruktionerna.

1.4 GPU

Vi kommer inte behöva någon enhet för att göra grafiska beräkningar, de kommer utföras av processorn. Vi behöver endast ett videominne som processorn kan skriva till och en VGA-motor som skriver videominnet till skärmen.

1.5 Programmering

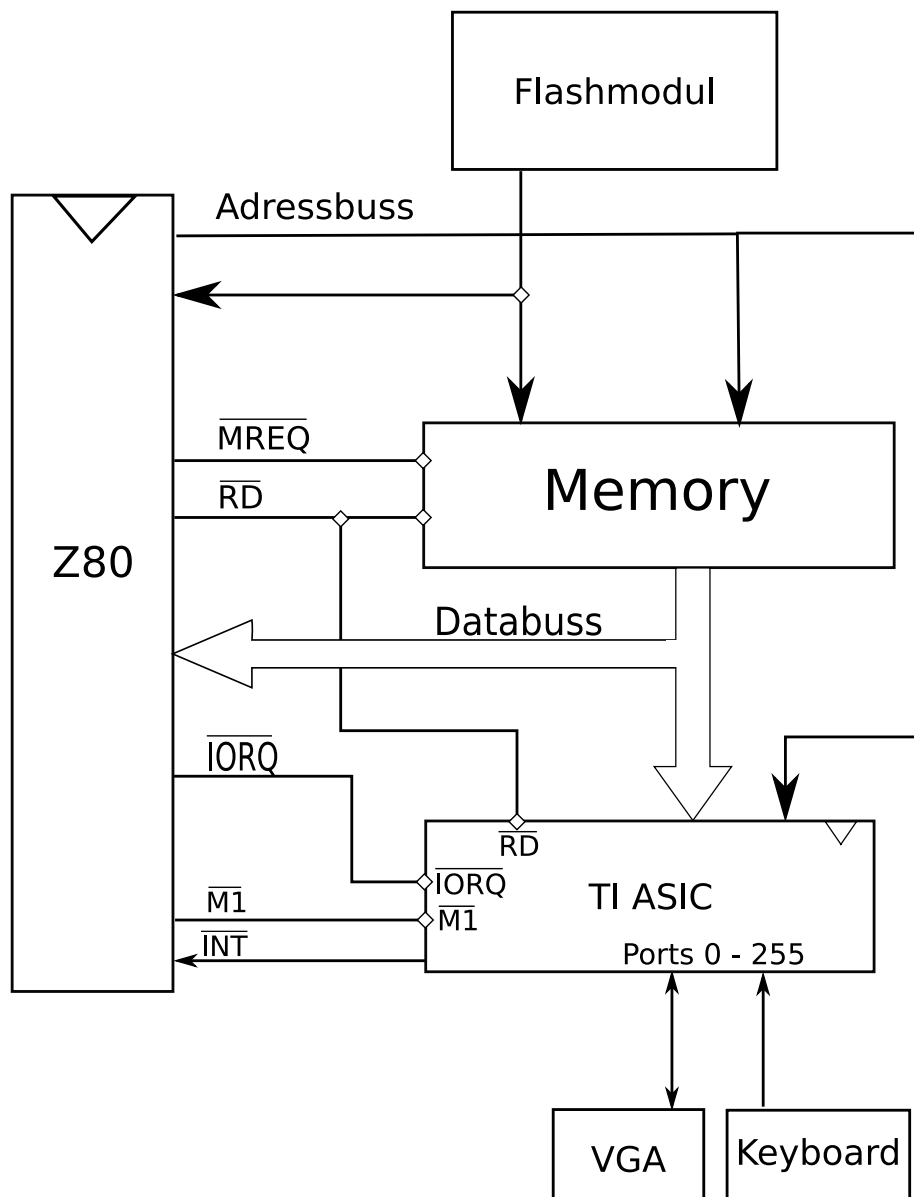
Z80:n programmeras med Z80:ns instruktionsset. Z80 är en populär processor med många verktyg tillgängliga. Vi kommer programmera och använda oss av tillgängliga assemblers för att skapa maskinkod.

2 Milstolpe

Vi har tänkt att göra klart z80:ns instruktioner (ej avbrott) samt koppla den till en VGA-motor som kan skriva registernas värden till VGA-skärmen.

3 Blockschema

3.1 TI84



3.2 Z80

