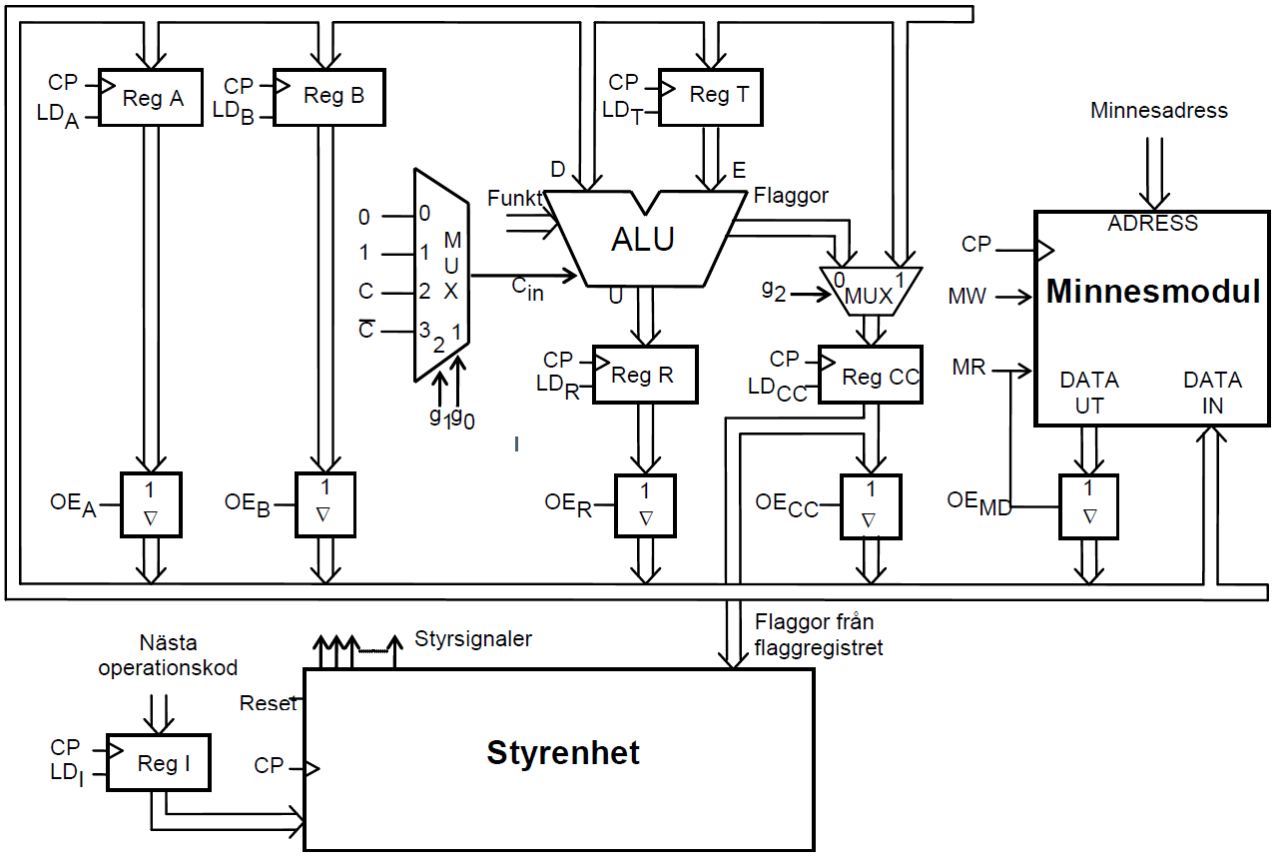
von Neumanndatorn (Ext-8)



Figur F.1 Databehandlande enhet med minne.

von Neumanns idé

"Man kan koda de olika operationerna som binära tal på samma sätt som data och lagra både operationer och data i minnet."

=

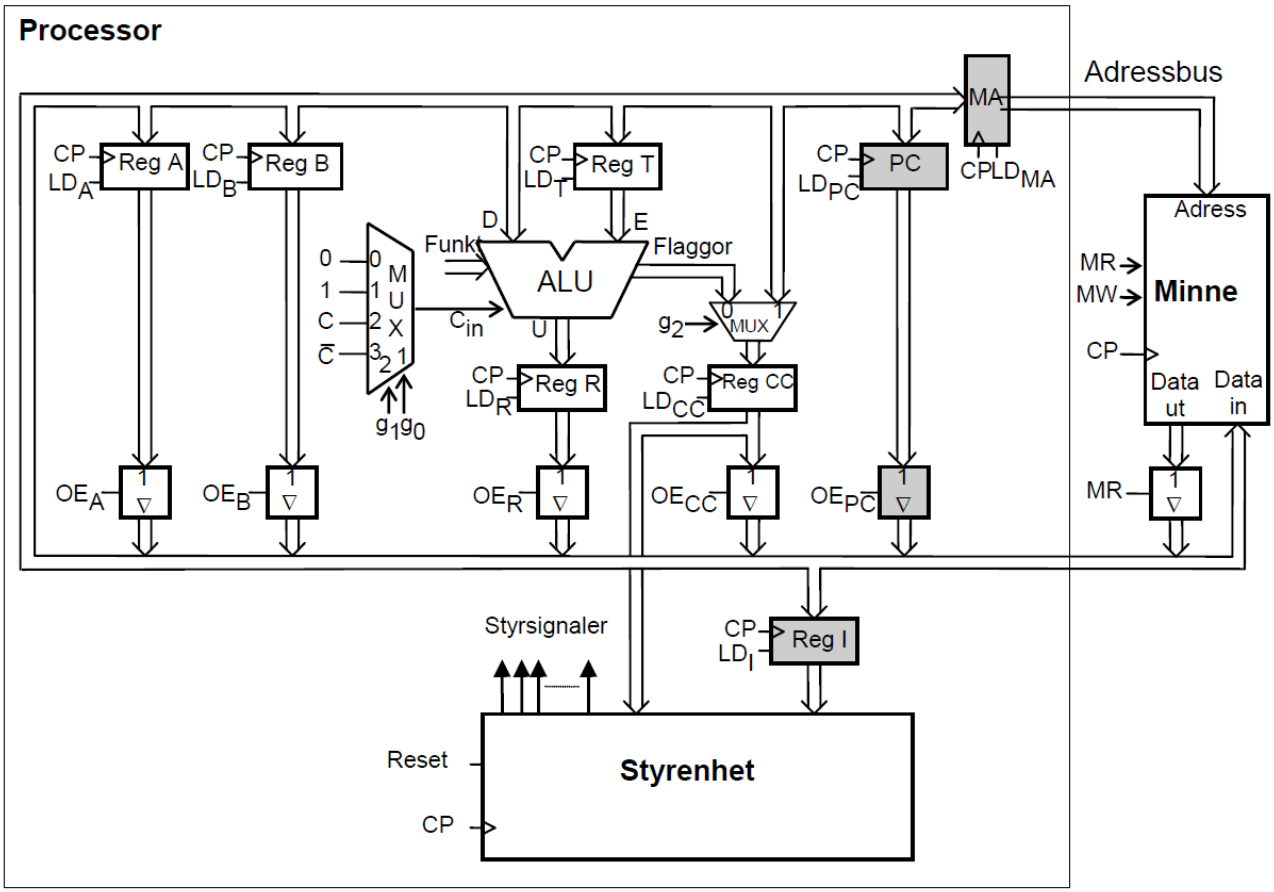
"Det lagrade programmets princip."

Data kan behandlas genom att en följd av instruktioner och data växelvis hämtas från minnet. Instruktionerna utför de önskade operationerna på data och skriver vid behov tillbaka data i minnet.

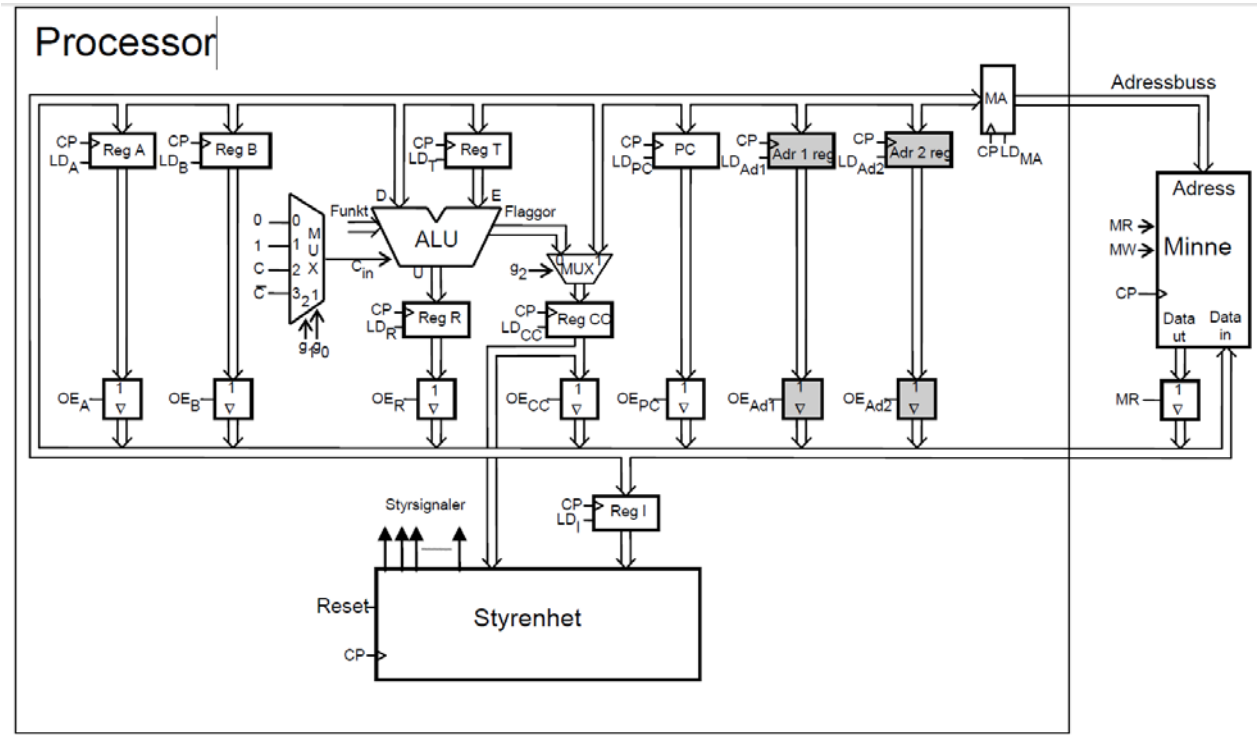
En instruktion är alltså kodad som ett binärt tal. En del av bitarna i detta tal är koden för en operation medan övriga bitar är en eller flera operander (data). En typisk instruktion består alltså av två delar enligt figuren nedan.

Operationskod	Operand(er)

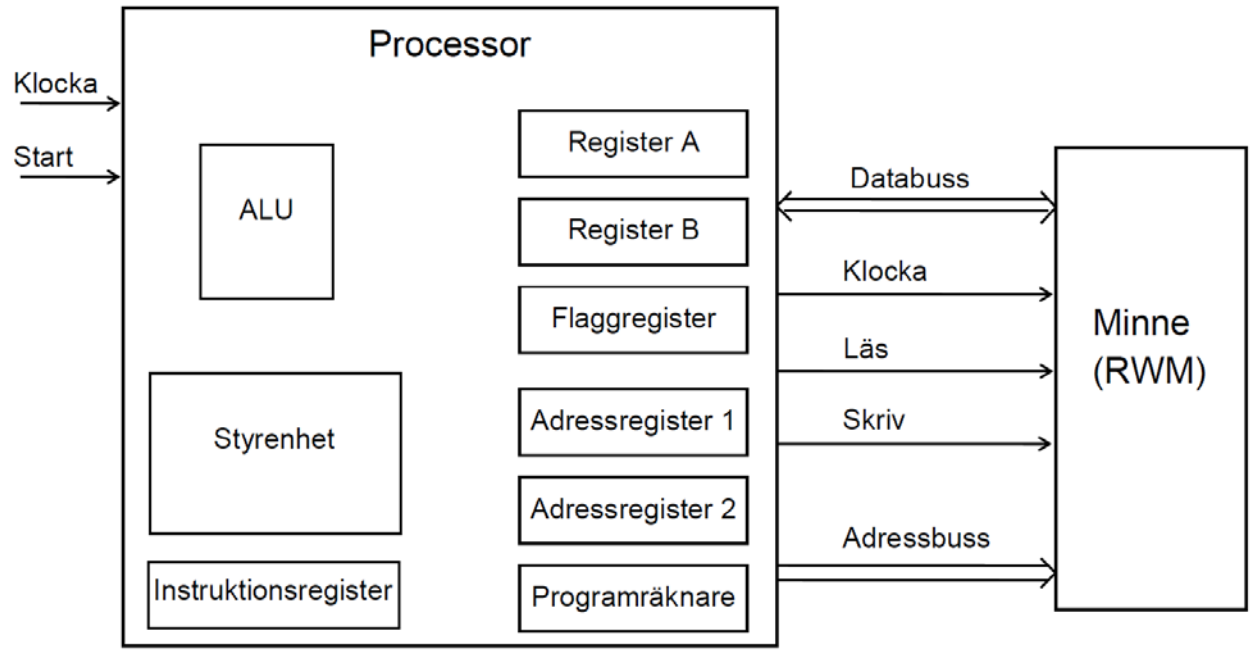
Operanden är antingen *data* (datavärdet) eller någon form av *adress till data*. Vissa instruktioner saknar dock operand och därmed också operanddel.



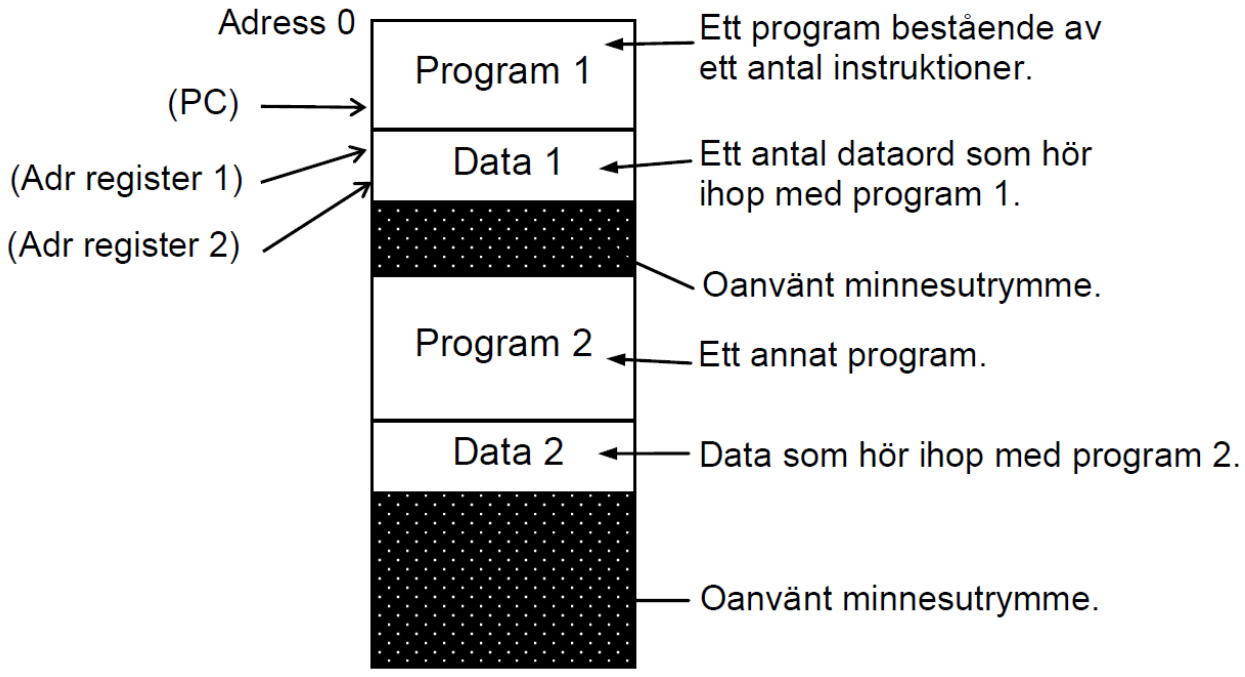
Figur F.2 Kombinationen von Neumannprocessor-minne.



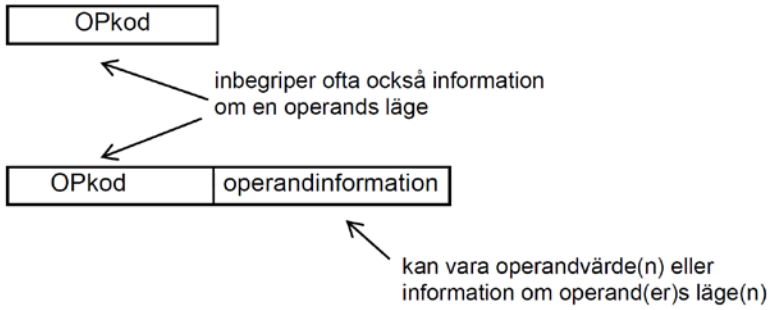
Figur F.3 Kombination av en utökad von Neumannprocessor och minne.



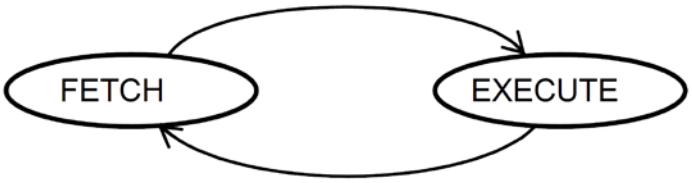
Figur F.4 Programmeringsmodell för enkel von Neumanndator.



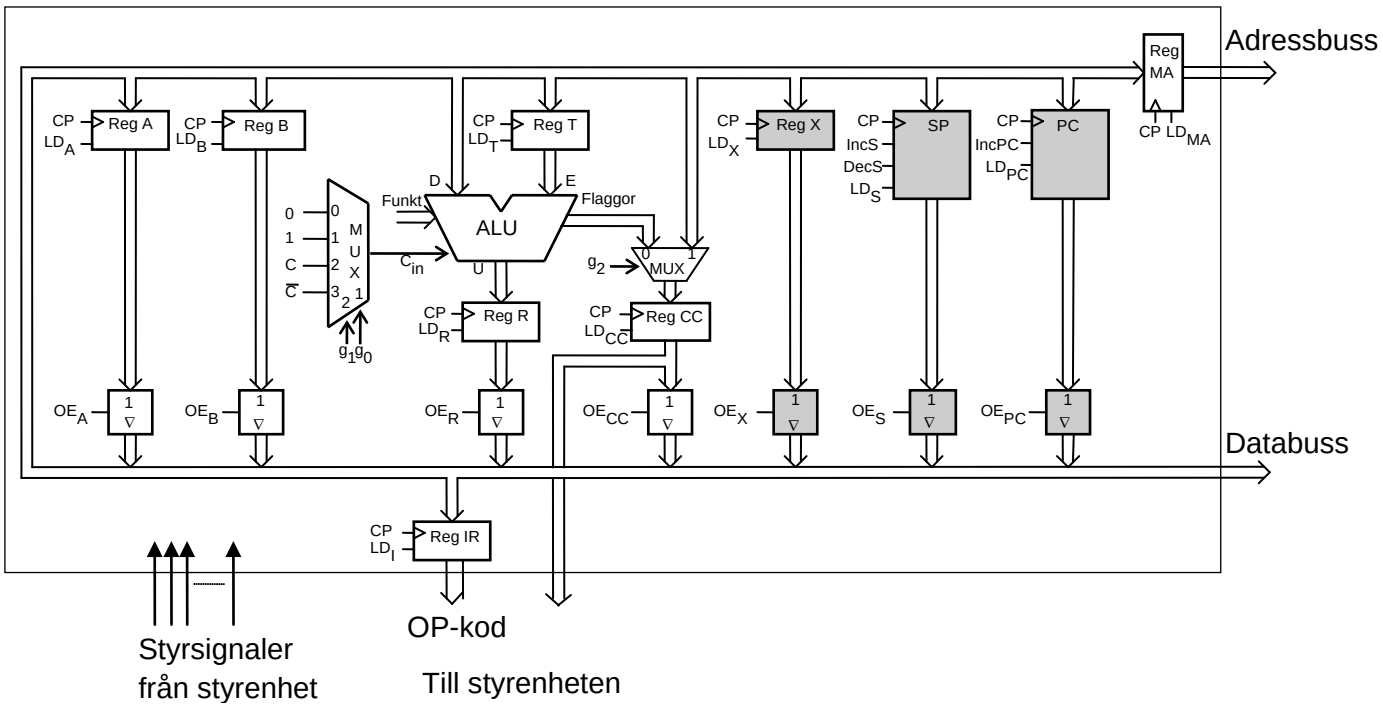
Figur F.5 Minnesdisposition för von Neumanndator.



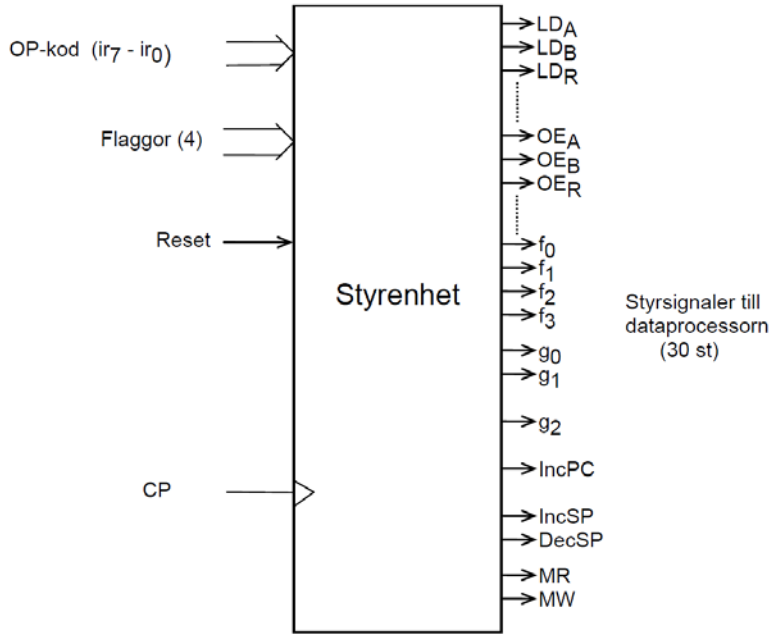
Figur F.6 Instruktionernas längd och innehåll följer ett sk instruktionsformat.



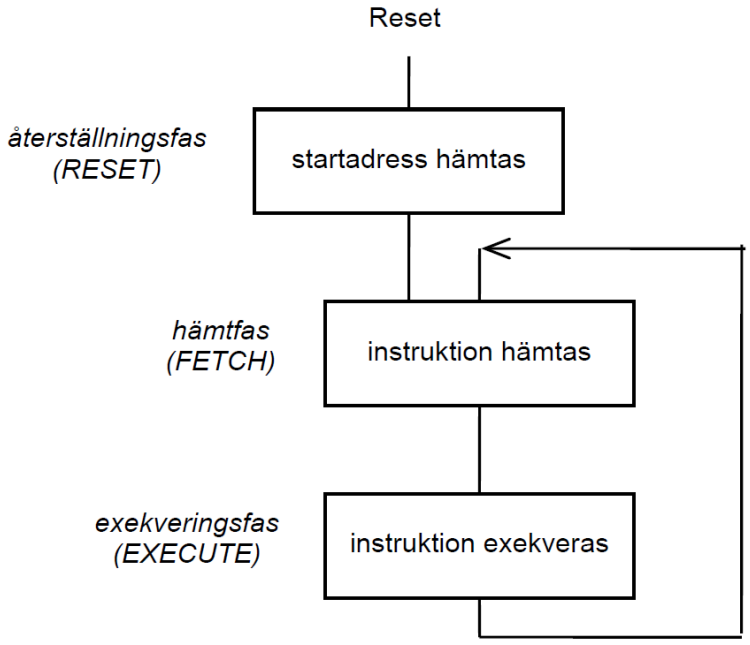
Figur F.7 Tillståndsgraf för instruktionernas två faser.



Figur F.8 FLEX-processorns dataväg.



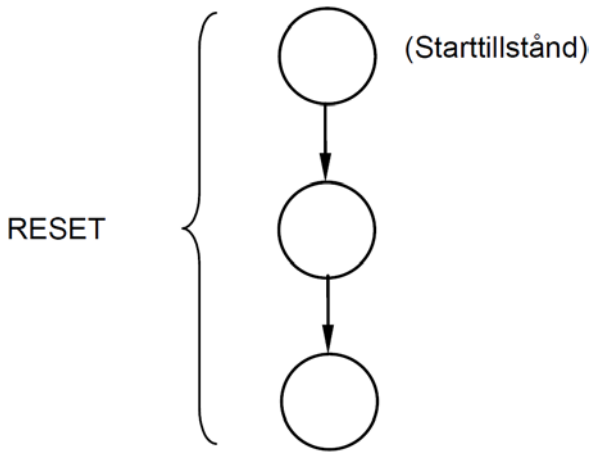
Figur F.9 FLEX-processorns styrenhet



Figur F.10 Instruktionscykeln kompletterad med s k återställningsfas.

Tabell F.1 Detaljerad beskrivning av RESET-fasen för FLEX-processorn

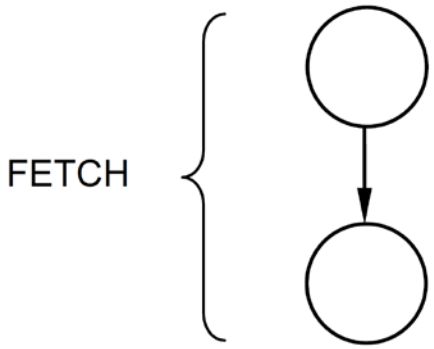
Klockcykel (State nr)	RTN-beskrivning	Styrsignaler	Kommentar
0	$FF_{16} \rightarrow R$	ALU-funktion = F_{16}, LD_R=1.	ALU-funktionen väljs så att talet FF_{16} finns på ALU:ns utgång. Laddingången på R-registret ettställs så att utvärdet från ALU'n (FF_{16}) laddas i R-registret vid nästa klockpuls.
1	$R \rightarrow MA$	OE_R=1 LD_{MA}=1.	Talet FF_{16} i R-registret kopplas ut på bussen. Talet FF_{16} på bussen laddas i minnesadress-registret vid nästa klockpuls.
2	$M \rightarrow PC$	MR=1, LD_{PC}=1.	Minnesinnehållet på adressen FF_{16} läses genom att minnet aktiveras för läsning. Det dataord som läses placeras i PC vid nästa klockpuls. Nästa klockcykel skall vara den första i fetchfasen.



Figur F.11

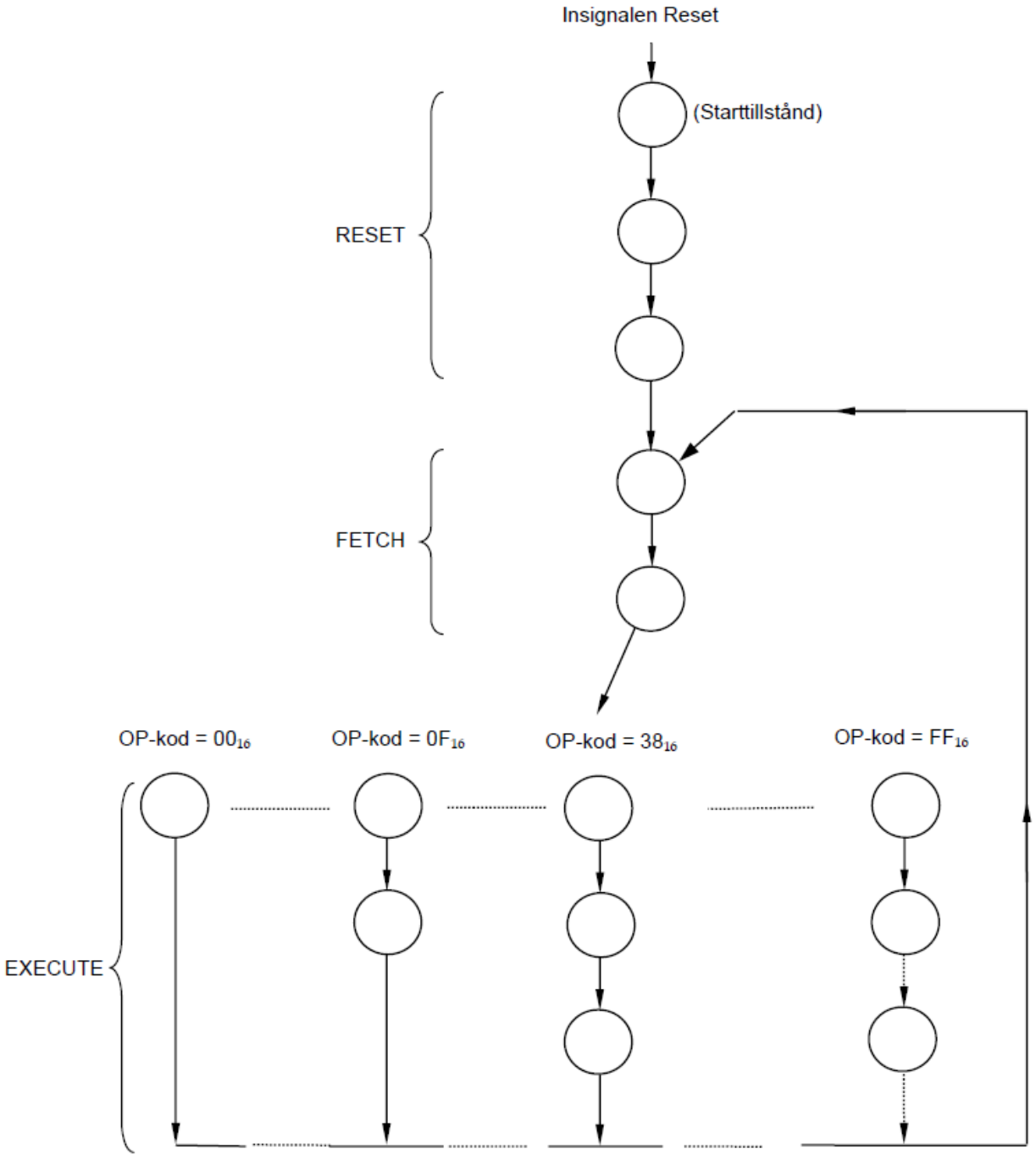
Tabell F.2 Detaljerad beskrivning av FETCH-fasen för FLEX-processorn

Klockcykel (State nr)	RTN-beskrivning	Styrsignaler	Kommentar
0	PC→MA, PC+1→PC	OE_{PC}=1, LD_{MA}=1, IncPC=1.	Adressen för nästa instruktions operationskod kopieras från PC till minnesadressregistret MA. Adressen som finns i PC ökas med ett.
1	M→IR	MR=1, LD_I=1.	Läs operationskoden från minnet. Placera den i instruktionsregistret IR. Nästa klockcykel skall vara den första i executefasen.



Figur F.12

Tillståndsgraf för FLEX-processorns arbetsfaser



Figur F.13 Tillståndsgraf för styrenheten till FLEX-processorn.