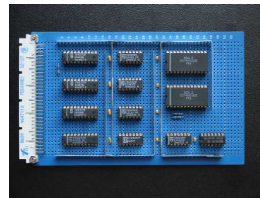


Digital- och datorteknik



Föreläsning #19

Biträdande professor Jan Jonsson

Institutionen för data- och informationsteknik
Chalmers tekniska högskola

Undantagshantering

Normaltillstånd vs undantagstillstånd

I normaltillstånd arbetar processorn i enlighet med vad vi hittills antagit, d v s den hämtar och utför instruktioner i en, potentiellt oändlig, repetitiv process.

För att kunna reagera på externa händelser, samt oväntade interna händelser, är processorn dessutom utrustad för att hantera undantag ("exceptions") från det normala beteendet.

Processorns mottaglighet för externa händelser gör det möjligt för användaren eller andra aktörer i datorsystemets omgivning att påverka processorns programflöde via speciella insignaler.

Processorns mottaglighet för interna händelser innebär att den kan reagera på onormala situationer i programexekveringen.

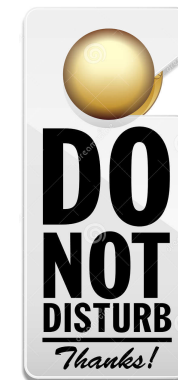
Undantagshantering

Normaltillstånd vs undantagstillstånd

I normaltillstånd arbetar processorn i enlighet med vad vi hittills antagit, d v s den hämtar och utför instruktioner i en, potentiellt oändlig, repetitiv process.

För att kunna reagera på externa händelser, samt oväntade interna händelser, är processorn dessutom utrustad för att hantera undantag ("exceptions") från det normala beteendet.

För att processorn inte skall påverkas av händelser som inte är kritiska finns det möjlighet att maskera vissa typer av undantag. Huruvida maskering skall ske eller ej kontrolleras genom speciella flaggbitar i processorn.



Undantagshantering

Typer av undantag – generellt för processorer

Återstart ("reset")

Händelse som alltid föranleder återstart av processorn.
Denna typ av händelse kan inte maskeras.

Avbrott ("interrupt")

Externa händelser som inte innebär att processorn skall återstartas. Denna typ av händelser kan vanligtvis maskeras.

Internt fel ("trap")

Händelser som uppträder vid utförandet av en instruktion och som gör att instruktionen inte kan fullföljas, t ex otillåten operationskod eller division med noll. Denna typ av händelser kan oftast inte maskeras.

Undantagshantering

Typer av undantag – FLIS-processorn

Återstart ("reset")

Aktiv signal på ingången RESET ger återstart av processorn.
Denna händelse kan inte maskeras.

Avbrott ("interrupt")

Aktiv signal på ingången IRQ ("interrupt request") begär ett avbrott hos processorn.
Denna händelse kan maskeras med processorns I-flagga.

Internt fel ("trap")

Utförandet av en instruktion med otillåten operationskod ger upphov till internt fel i processorn.
Denna händelse kan inte maskeras.

Undantagshantering

Hanteringsrutiner – FLIS-processorn

Återstart ("reset")

Adressen till huvudprogrammets början hämtas från datorsystemets resetvektor (= innehållet i adress \$FF).

Avbrott ("interrupt")

Adressen till en hanteringsrutin ("interrupt handler") hämtas från datorsystemets avbrottsvektor för extern styrning (= innehållet i adress \$FD).

Internt fel ("trap")

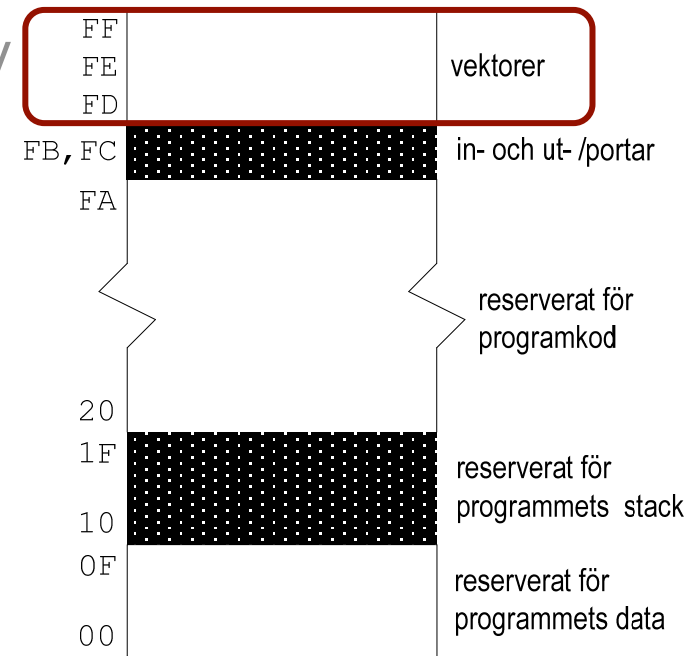
Adressen till en hanteringsrutin ("trap handler") hämtas från datorsystemets avbrottsvektor för internt fel (= innehållet i adress \$FE).

Undantagshantering

Primärminnets användning

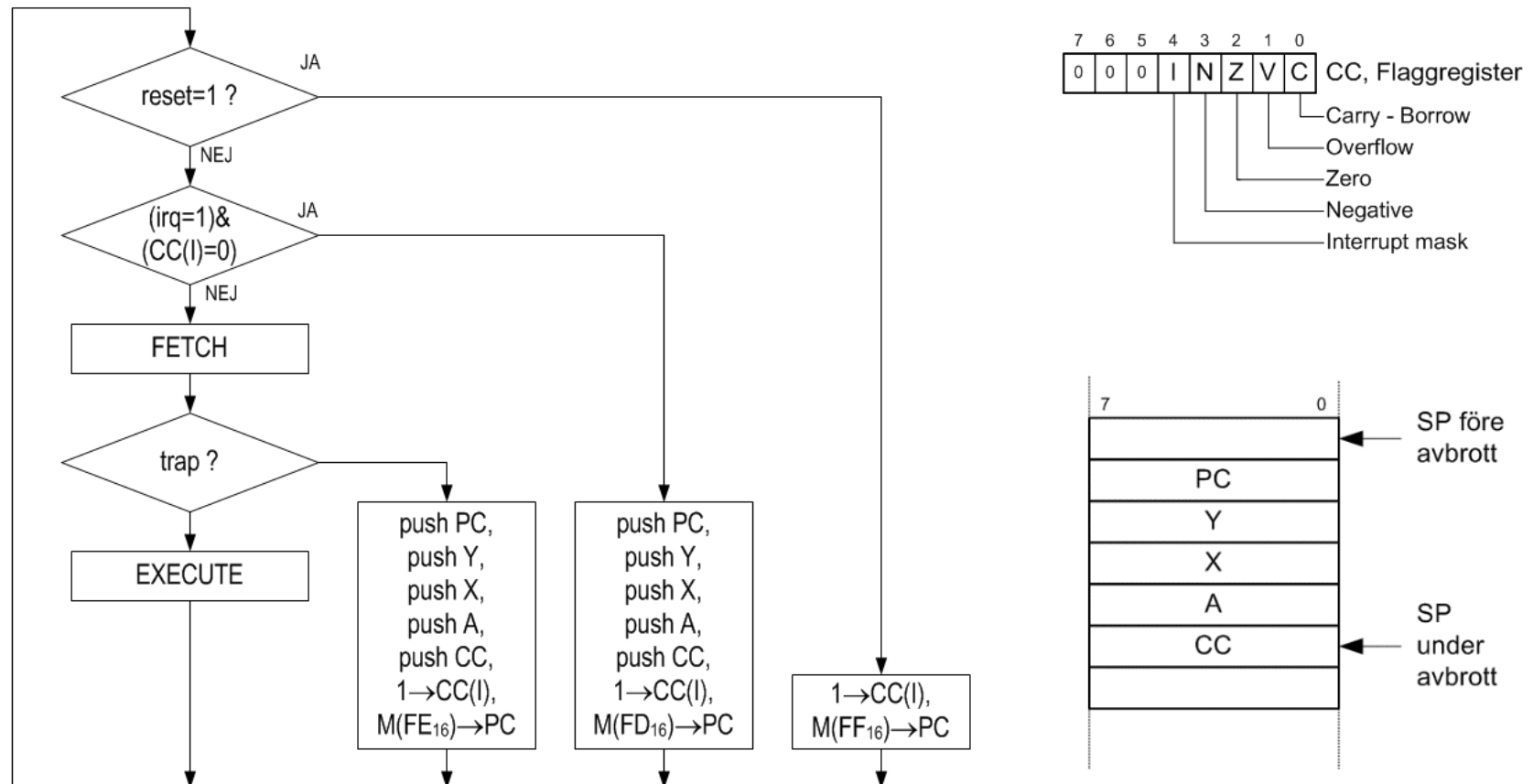
Den rekommenderade användningen av primärminnet i FLIS-processorn är:

- Adress 20_{16} – FA_{16} : Programkod
- Adress 00_{16} – $0F_{16}$: Globala variabler (data med relativt lång livslängd)
- Adress 10_{16} – $1F_{16}$: Programstack (data med relativt kort livslängd)
- Adress FB_{16} – FC_{16} : In- och ut-portar (reserverat för kommunikation med sensor och/eller ställdon)
- Adress FD_{16} – FF_{16} : **Vektorer** (reserverat för lagring av speciella hoppadresser)



Undantagshantering

Undantagshantering – FLIS-processorn



Undantagshantering

Återstart – FLIS-processorn

- När signalen på RESET-ingången blir aktiv upphör processorns arbete.
- Processorn inväntar att ingången blir inaktiv.
- I-flaggan ettställs för att maskera (d v s, stänga av) avbrottssystemet.
- Processorn läser innehållet i datorsystemets RESET-vektor på minnesadress FF_{16} , och placerar det i register PC.
- Processorn hämtar första instruktionen i huvudprogrammet.

Undantagshantering

Avbrott – FLIS-processorn

- När signalen på IRQ-ingången blir aktiv exekverar processorn färdigt den instruktion den håller på med.
- Om avbrottssystemet är aktiverat, d v s I-flaggan är nollställd, accepterar processorn avbrottet. I annat fall fortsätter programexekveringen som vanligt.
- Om avbrottsbegäran accepteras sparas innehållen i de interna registren i ordningen PC, Y, X, A och CC på stacken. Därefter ettställs I-flaggan för att utestänga nya avbrott.
- Processorn läser innehållet i datorsystemets IRQ-vektor på minnesadress FD_{16} , och placerar det i register PC.
- Processorn hämtar första instruktionen i hanteringsrutinen.

Undantagshantering

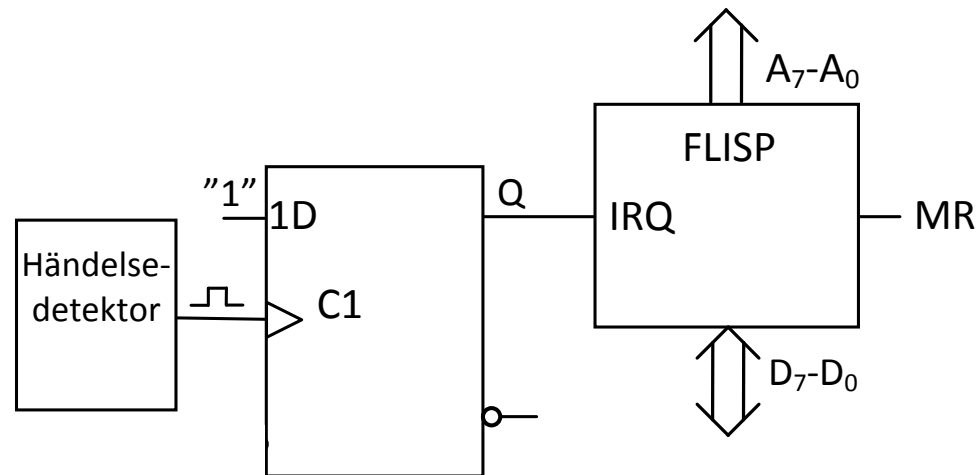
Internt fel – FLIS-processorn

- Om processorn identifierar en otillåten operationskod sparas innehållen i de interna registren i ordningen PC, Y, X, A och CC på stacken. Därefter ettställs I-flaggan för att utestänga ytterligare avbrott.
- Processorn läser innehållet i datorsystemets TRAP-vektor på minnesadress FE_{16} , och placerar det i register PC.
- Processorn hämtar första instruktionen i hanteringsrutinen.

Undantagshantering

Detektering av extern avbrottssignal

Med hjälp av en D-vippa kan såväl kortvariga som långvariga händelsesignaler registreras av processorn.



Undantagshantering

Kodning av hanteringsrutiner

När hanteringsrutinen börjar exekveras vet man att händelsen som orsakar avbrott har inträffat. Hanteringsrutinen skall därför utföra den kod som är förknippat med händelsen och sedan återvända till det avbrutna programmet med återställda register.

- Hanteringsrutinen skall byggas på samma sätt som en vanlig subrutin, med följande viktiga skillnad: hanteringsrutinen ska avslutas med instruktionen RTI ("return från interrupt"), som återställer innehållen i samtliga processorregister.
- Vid avbrott på grund av extern händelse måste den aktiva signal som orsakade det pågående avbrottet ha försvunnit innan återhoppet. Annars kommer ju ett nytt avbrott att genereras för samma externa händelse efter återhoppet.

Undantagshantering

Kodning av hanteringsrutiner

Återhopp från avbrott görs med instruktionen RTI.

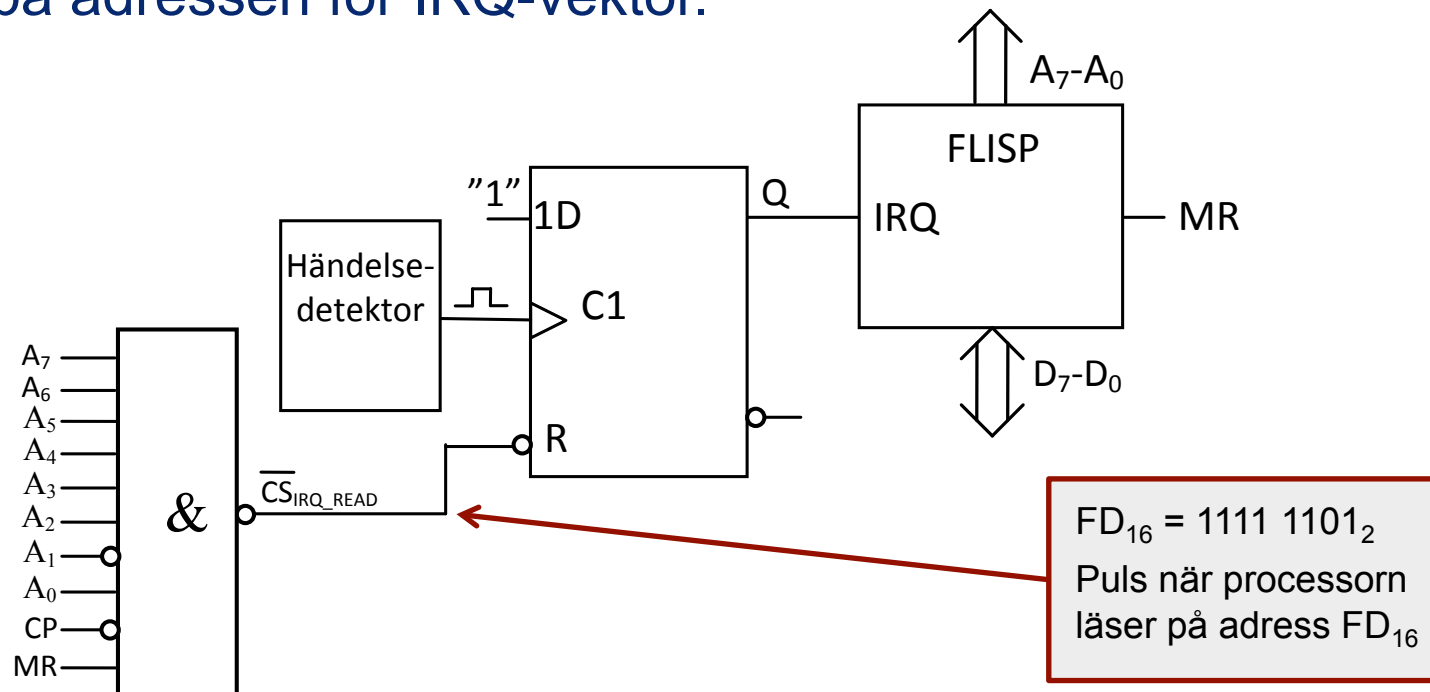
Notera att avbrottssystemet automatiskt aktiveras då de gamla registerinnehållen i processorn återställs vid återhopp från hanteringsrutinen eftersom det gamla CC-innehållet, som hämtas från stacken, innehåller en nolla i I-biten.

Instruktion		Adressering			Operations- beskrivning	Flaggor				
Operation	Beteckning	Inherent				4	3	2	1	0
		OP	#	~	I	N	Z	V	C	
Return from interrupt	RTI	44	1	6	M(SP) → CC; SP+1 → SP	Δ	Δ	Δ	Δ	Δ
					M(SP) → A; SP+1 → SP					
					M(SP) → X; SP+1 → SP					
					M(SP) → Y; SP+1 → SP					
					M(SP) → PC; SP+1 → SP					

Undantagshantering

Kodning av hanteringsrutiner

Nollställning av den signal som genererat avbrott p g a extern händelse kan ske genom att läsa på adressen för IRQ-vektor.



Undantagshantering

Demonstrationsexempel – tongenerator

Du skall skriva ett assemblerprogram som genererar en ton (fyrkantvåg) med frekvensen 250 Hz på bit b_0 på utport FB_{16} . Tonen skall genereras under ett tidsintervall av 180 ms.

- a) Skriv programmet så att alla nödvändiga fördröjningar erhålls genom att maskinkod exekveras. Antag att FLIS-processorn har klockfrekvensen 1 MHz.
- b) Modifiera programmet så att det istället utnyttjar att en extern pulsgenerator är ansluten till IRQ-ingången. Antag att pulsgeneratoren levererar positiva klockflanker en gång per millisekund.