

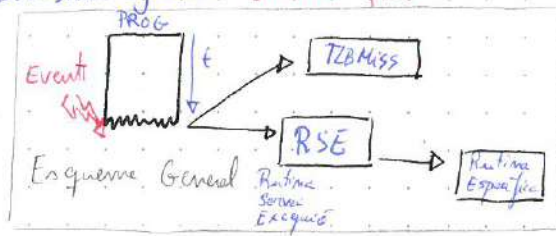
EXCEPCIONS I INTERRUPCIONS

Introducció

Excepcions són mecanismes de la CPU d'alterar flux de control sense instruccions de salt.

Excepcions Comunes: a) Error Intern $\begin{cases} \text{Accés no alineat} \\ \text{Division by zero} \end{cases}$ ∇ **NOMÉS float**

Interrupcions són generades per E/S.
Interrupcions & Excepcions:
b) [TLB/TP] miss
c) Syscalls i traps
d) Interrupcions



Loprocessador Del Sistema (CPO)

Part física de la CPU amb banc de registres específics.

EPC (\$14) $\begin{cases} \text{PC Inst.} \end{cases}$ Còpia el PC SEMPRE. Interrupció: $EPC \leftarrow PC + 4$
Altres: $EPC \leftarrow PC$

Status (\$12) $\begin{cases} 15 \\ 8 \\ 1 \end{cases}$ $\begin{cases} \text{EXL: Executació Level (Mode Execució)} \end{cases}$ $\begin{cases} 1 \equiv \text{Mode Kernel} \\ 0 \equiv \text{Mode Usuari} \end{cases}$

Cause (\$13) $\begin{cases} 15 \\ 8 \\ 6 \\ 2 \end{cases}$ $\begin{cases} \text{IM: Interrupt Mask. Determina si val interrupció d'un grup específic.} \end{cases}$

Podem pensar que
Status \equiv "Config"
Cause \equiv "Injunció"
 $\begin{cases} \text{ExcCode: Número Excepció} \\ \text{IP: Senyal de petició d'interrupció d'un grup específic.} \end{cases}$

Exception Codes (Alguns)

0 \rightarrow (Int) Interrupció

1 \rightarrow (Mod) Miss de TLB amb D=1. "Primera excepció a TLB i exitòs que D=1"

2 \rightarrow (TLBL) Miss [TLB/TP] Lecture
3 \rightarrow (TLBS) Miss [TLB/TP] Escriptura
 $\left. \begin{array}{l} 2 \\ 3 \end{array} \right\}$ Com que hi ha rutines només per "TLB Miss" el HW decideix a quin cridar i si es crida a "RSE" (per descartar) sig. que es tracta de TPmiss.

8 \rightarrow (Sys) Syscall (Causat per instrucció "syscall")

13 \rightarrow (Tr) Trap (Causat per instrucció "trap").

PAS 1: Hardware

① Aturen l'execució $\begin{cases} \text{PC} \\ \text{PC+4 (Interrupció)} \end{cases}$

② Guardar PC en "EPC"

③ Possem en "ExcCode" el valor que petiqui. ∇ Hi ha prioritats: Totes > Interrupcions.

④ Fiquem mode Kernel EXL=1. ∇ Durant mode Kernel les interrupcions Inhibides.

⑤ Guardem al PC la direcció de la rutina que toqui. $\begin{cases} \&RSE \\ \&TLB Miss \end{cases}$ $\left. \begin{array}{l} \text{Aguanta \& senyal} \\ \text{amb bit 31=1 Pq} \\ \text{són de S.O.} \end{array} \right\}$

PAS 2: Software (RSE)

① Preservar Context. Guardem tots els registres menys \swarrow $\$K0, \$K1$ # Registres del S.O.
 \searrow $\$fX$

② Identificar excepció i ubicar la subrutina. Això es coneix com "Exception Handler".

③ Guiar a subrutina específica. Pot:

- Avortar.
- Guiar a un altre programa.
- Resoldre i retornar.

④ Quan acaba la subrutina el RSE ha de considerar ajustar EPC. Sumar 4 si era syscall.

⑤ Restaurar context.

⑥ Retornar al programa amb instrucció "eret" que fa:

- Canviar $EXL = 0$

- $PC \leftarrow EPC$ (En cas de que hagi de retornar).

Si no fes això entraria en un bucle infinit de cridades al syscall.