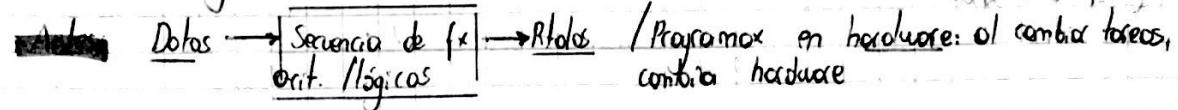


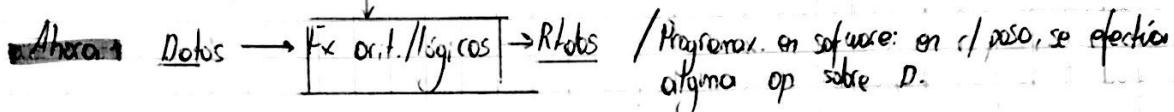
Arquitectura de computadoras

Close 1

Concepto de programa



Intérprete de inst + cod. inst + señales de control → implementación



- P/ cl. paso se necesita un nuevo conjunto de señales de control
- Las inst. proporcionan otras señales
- Nuevo concepto de programación

Arquitectura von Neumann

- CPU formada por UC y ALU
- Datos e inst. deben introducirse en el sist. y rotos se proporcionarán el componentes E/S
- Necesario el manejo temporal D e instrucciones en memoria ppol.
- PC: contador de program
- IB: reg. de instx
- MAR: reg. dir. mem.
- MBR: reg. buffer mem.
- F/S AR: reg. dirx de E/S
- F/S BR: reg. buffer de E/S.

Repertorio de instrucciones

Conjunto completo de instrucciones que se realizan en una CPU.

Representado x un conj. de código de ensamblaje.

De op: ADD, SUB, LOAD

De operando: ADD, BX, PEPE

Elem. de una instrucción

- Cod. operax
- Ref o op fuentes
- " rotos
- " sig. instrux.

Almacenamiento de operandos

- Mem. ppol
- Mem. virtual o mem. cache
- Reg. CPU
- Dispositivo E/S

Tipos de instr.

- Procesamiento de D
- Inst. aritmético-lsg.
- Almac. de D
- Inst. de memoria
- Transf. de D
- Inst. de E/S
- Control
- Inst. de testeo y flujo.

Cant. direc

+ direc x inst: inst + complejos, + reg., - inst. en el program.

- " : inst. - complejos, + inst. en program., ejecuc de inst. + rápida

Decisiones en el diseño del conjunto de instrucciones

Tipos de operandos

- Repertorio de op
- Cuotas, cuotas, complejidad
- Modos de direccionamiento

Formato: longitud, nro. direc, formato comp.

Registros: nro. reg. referenciables.

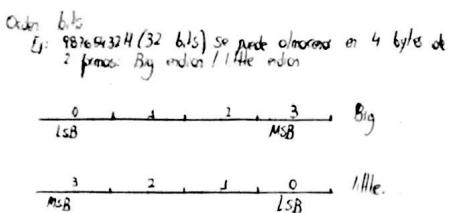
RISC contra puesto a CISC → 2 f.p.s de org.

• Computadora de conj. reducido de instx

• Computadora de conj. complejo de instx.

Tipos de operando

- Aleatorios
- Nros.
- PF, ptos flotante
- Cadetes
- ASCII, EBCDIC, etc.
- Datos logicos
- Bits (reg. flags)



Big endian: el byte + significativo en 10 direcs cf. valor num + bajo

Dir byte	Furm 1	Furm 2
00	98	32
01	76	54
02	54	76
03	32	98

Little endian: el byte - significativo en 10 direcs cf. valor num + bajo

Tipos de operaciones

1. Transf D
2. Aritméticos
3. Logico
4. E/S
5. Control del sist.

1) Efectos: ubicar op fuente y dest / tomar D / modd.
Dif. mor - dif. inst.

- 2) Op. básicas (sum, sub o sin signo o PE) u otras op. como inc, dec, neg, absolute, shift left/right.
- 3) Op. q: manipulan bits (booleanas, rotate left/right) / op. p: combinar fuentes (ASCII > EBCDIC)
- 4) IN/OUT - MOV. Se pide hacer mediante un controlador o parte.

- 5) Modifican el valor contenido en PC
- Salto condicional - JMP
 - Salto cf. retorno o llamada o subrutina - CALL/RET
 - Salto condicional - JZ

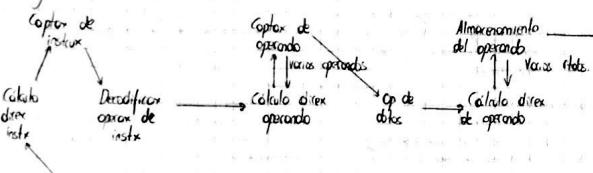
MOD

- Inmediato
- Directo de mem. /absoluto.
- Directo de reg.

Ciclo de instrucción

- Ciclo de copias: lo directo de inst. q se debe copiar se encuentra en PC. La UC copia lo inst. desde memoria y va al IR. PC incrementa. UC interpreta inst. y lo lanza o retira
- Ciclo ejecución: opciones posibles
- Acceso dir. memoria
 - Transf. D. de CPU-Memoria
 - Proc. - E/S
 - Transf. D. CPU y módulo E/S
 - Algunas combinaciones de todos.
 - Algunas combinaciones de todos.
 - Acceso memoria D
 - Op. aritmética
 - Control
 - Algunas secuencias -inst. desolto.

Diagrama de estados



Struktuos

Iniciar en longjmp de programación. Puede interrumpirse desde cualquier punto del programa. Brinda economía (puedes usar valores veras) y modularidad. Requiere paquete de paquetes.

Acceso de parámetros

- Va registros el nro. de reg es lo más limitado, se debe documentar reg. usados.
- Va memoria: usa área definida de memoria, difícil de estandarizar.
- Va pila: más fácil + usado. Independiente a memoria y registros.

Ejecución de una pila

El operando se encuentra en la cabeza de la pila. Requiere un registro Puntero de Pila (SP). Se pone q contiene la dirección de la cabeza de la pila.

Op. push sobre la pila: push oplo
pop desapila son inversas e/s

Operaciones de apilar/desapilar

Sacar el q de 2 operaciones. mor de datos Reg-Mem o Mem-Reg

q se modifica al punto de apilar/desapilar después de la anterior

! tener en cuenta donde apunta el puntero / como crece la pila

Apilax del procedimiento

nombre proc

... RET

nombre epip

Llamada al proc.

push q

push parámetros

CALL nombre

ORG 1000H

MOV BX, 0

Mov AX, dato

PUSH AX

CALL sub

ADD BX

HLT

BB 55H

END

Possibles pasos q en un procedimiento

- 1) Salvar BP (BP rego)
- 2) " SP
- 3) Asignar espacio p/ D. locales (optional)
- 4) Salvar val de otros reg ("")
- 5) Acceder a parámetros
- 6) Escribir sentencias o ejecutar
- 7) Retornar parámetros ("")
- 8) Regresar al procedimiento

org 300H dato

1) Poses
Conezará c/ push BD
mov BD, SP

Establecer a BD como punto de referencia y se usa el ordenador a parámetros y D de pl. SP no se puede usar. SP puede combinar para BD permanece quieto.

3) explanal

Se decrementa SP, reservando lugar sub, SP, 2
(reserva 2 bytes pt D/variables)

5) 2 (tamaño de BP oiplado); tamaño de direct de refamo + total parámetros del buceo y BP.

Solida del procedimiento
Los reg solados en lo piso deben ser descargados inverso. Si se resarri expirio p/
variables locales se debe regar so el volad de BRI q no cambie en per.

Report by
Volker of program d AE

4) optional
Ej: push DI
si el proc. no cambia val. de reg. no
necesita ser salvado.

11 fin mas x 88 se usa BX en vez de BD

Bibliografia W. Stollings: cap. 9/50 / opòdice 9A / cap. 3 / 11 / 50.

Práctica 1	Instrucción	Valor SP	Ax	Bx	2)	Instrucción	Valor SP
1)	mov 4x,5	8000 8	5			sg zooth	FFFF 61
	mov bx,3	8000 4	2			rutina mov bx,3	FFFF 60
	PUSH AX	2FFF 9	5			rot	FFFF 61
	PUSH BX	2FFF C	5			org zooth	FFFF 61
	PUSH BX	2FFF D	5			push ox	FFFF 61
	POP BX	3FFF 9	5			call rutina	FFFF 60
	POP BX	3FFF C	5			pop bx	8000 41
	POP 4x	8000 F	5			hlt	8000 40

3) a) rut
 org 300ch
 mov bx, 3
 ret
 org 200ch
 call rut 2002
 odd cx, 5
 call rut 2006
 hlt
 end

b) 1º cap est lo piq mada s sra de sra q
mata fentat al merca s sra s mada s

2º rebanc el pomo s sra s sra s

	Código	Registro	Pila	Volver	Referencia
4)	mov bx, 5 call subrutina mov dx, offset call subrutina				
	mov bx, 5 push bx call subrutina pop bx				
	mov cx, offset push cx call subrutina pop cx				
	mov dl, 5 call subrutina				
	call subrutina mov A, dx				

$$5) \text{ Colgalo } A+B-C$$

o) $\text{D}_2 + \text{H}_2\text{O} \rightarrow \text{D}_2\text{O} + \text{H}_2$

b) $\text{CH}_3\text{OH} + \text{H}_2\text{O} \xrightarrow{\text{acid}} \text{CH}_3\text{COOH} + \text{H}_2$

c) $\text{CH}_3\text{CH}_2\text{OH} + \text{H}_2\text{O} \xrightarrow{\text{acid}} \text{CH}_3\text{CH}_2\text{COOH} + \text{H}_2$

d) $\text{CH}_3\text{CH}_2\text{OH} + \text{H}_2\text{O} \xrightarrow{\text{acid}} \text{CH}_3\text{CH}_2\text{COOH} + \text{H}_2$

e) $\text{CH}_3\text{CH}_2\text{OH} + \text{H}_2\text{O} \xrightarrow{\text{acid}} \text{CH}_3\text{CH}_2\text{COOH} + \text{H}_2$

C. P. 2 sub 1, C sub D, C
D. PW no 5 Av. F.
avg 2000' 60' *avg 2000'*
avg 2000' 60' *avg 2000'*

act. A - B	full ratio	act. B - C
act. C	1.0	act. B + C
act. B - C	0.69	act. C
act. C	0.69	act. C - D
act. D	0.69	act. C - D

b) $\text{avg. } \text{Zec}^{(k)} = \frac{\text{avg. } \text{Zec}^{(k)}}{\text{avg. } B_x, C}$

END

A horizontal row of spiral notebook rings, showing the metal spiral binding mechanism.

۱۱۰

046 20004
Nov 11, 1993
S. 56 5pm?

192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210

c) ~~666,100~~

Close 2 • Mecanismo mediante el cual se puede interrumpir el procesamiento normal del PV. q' tiene una ejecución secundaria de instrucciones.

La interrupción puede ser de oxigeno interno o externo al CPU

Métodos de interrupciones - Atend. de uno ejecut. de una instruc. (Ej: V o ddr)

- Temporizador interno del procesador (permite al SO hacer $\frac{1}{x}$ tareas)
- CPU de ES/3 (pl. indica la finalizac. de una op.)
- Fondo de hardware (Error perdida de memoria, pérdida de energía).

• ¿Qué hacer si interrumpe?
 En los todos los casos implica transferir el control a otro programa (hostor) que:
 • Solve E procesoab
 • Corrige /responda malo q' ocurrió la interrup
 • Restore E original proceso.
 • Activa ejecución normal program interrumpido.

Jerarquía de interacciones
si hay múltiples que pueden solicitar interrupciones se establecerán cuáles son más importantes. Se consideran:

- No emascarables: no pueden ignorarse. Son eventos peligrosos o de l' prioridad
- Emascarables: pueden ser ignorados. c/ inst. poderosa inhibir lo solicitado

Interrupciones por hardware generadas por dispositivos de E/S. Son "verdaderas" interrupciones. El sistema de cómputo tiene q' manejar eventos ext. "no planeados" o "asincrónicos". No se detienen los procesos en ejecución en ese momento.

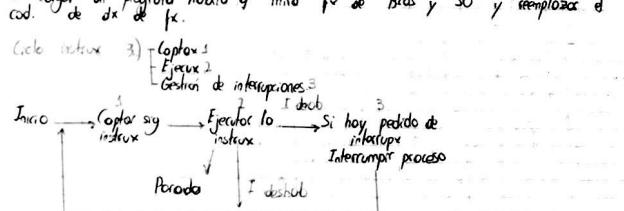
They tried and have the impression

Vector analysis problem

Interacciones < hardware creados < el propósito en ito. o ciertas áreas:
- Con empresas: V en ALU de pto flotante.
- Fallo del program: Falta de ejecutar instr. no definido.
- de hardware: error poridad de memoria.
- Acceso no threads o a zonas de memoria protegidas.

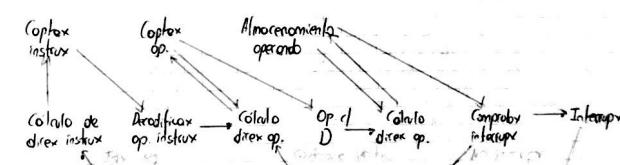
Interac x sefones: Muchos proc. tienen nsls espíritus q' afectan al procesado de la misma materia q' los "Grol" o "cucos". No requieren pl horas llamadas concia direk de la futura en tiempo de ejecu.

Sí no tuviera interrupciones por software, debería escribir todos los fx g/ negativo) o
si creara un documento en la memoria flash de la cámara.



Caso de Interrup.

- Se comprueba si se ha solicitado alguna interrupc. (Indicador x flag) → interrup.
- Si no hay señal, se copia lo sig. instrucción
- Si hoy algún pedido de interrupc. pendiente:
 - Se suspende ejecución en curso
 - Guarda contexto (prox. instrucción ejecutar y E procesador). Flag de ejec.
 - Coloca el PC al direcc. romana o rutina de gestión
 - Finalizada la rut. de gestión, el prox. retorno lo ejecuta del program.



(1) Adm. de l'Int. PIC

Interrupciones múltiples

Interrupciones inhabilitadas
El procesador puede y debe ignorar la señal de petición de interrupción si se produce una interrupción en ese momento.
Si se habilita generando una interrupción, se mantiene pendiente y se examinará luego una vez que se hayan habilitado más.
Ocasión una interrupción, se inhabilita; se gestiona la misma y luego se habilita otra vez.
En lo todo, las interrupciones se manejan en un orden jerárquico estricto.

Definir prioridad
una interrupción de prioridad + 1 puede interrumpir a un gestor de interrupción de prioridad x
Cuando se ha gestionado la interrupción de prioridad + 1, el procesador vuelve a las interrupciones de prioridad x.
Terminados los rutinos de gestión, se retorna al programa del usuario.

Reservación de interrupciones

Interrupciones multihilos
Un dispositivo g que provoca interrupciones tiene una entrada física de interrupción conectada al CAN. Servicio poco común.

Línea de interrupción única

Una sola entrada física de pedido de interrupción o la que están conectadas los dispositivos se dice "pregatados" a el disp. Si ha producido el pedido de interrupción "polling/muestreo".

Interrupciones vectorizadas.
El disp. q. quiere interrumpir ademas de la señal de pedido de interrupción, debe codificar en el bus de D su identificador.
Lo hace el periférico dirigido o controlador de interrupción, que no existe.

Escenario de trabajo

Si el procesador tiene una única F de pedido de interrupciones. (Adm. de l'Int. PIC)
Si en cambio varios proveedores de interrupciones se solucionan vía el PIC "dispositivo controlador programable de interrupciones".

Interrupciones del MSX80
Hardware: línea INT (lado de conocimiento INTA), /línea NMI (proces. de errores x solo int.).
Software: instr. INT xx. // el retorno desde el gestor debe usarse la instr. RET.

Proceso de orden vectorizado.

Tabla de UC de interrupciones

Es el espacio reservado de tipo de interrupción (0..255) y el proc. designado p/ atenderlo.
c/ E. es una tabla doble (4 bytes).

Dirección del proc. q. brinda el servicio.

UC preasignados:
0= finalización ejecución
3= ploto de escrito
6= lectura entrada. Requiere BX
7= escritura de S. y AL

• Se pide en ploto de PIC, horario e int. de ordenador apagado

PIC: programable para el ploto de ordenador que tiene memoria.

Cortejada de interrupciones?

Registros internos PIC *

Configuración

Los reg. q. del PIC se sitúan desde 20h. Se accede c/ op de lect. y esc.

Interrupciones de hardware asignadas:

INT0 - F00
INT1 - timer
INT2 - handshake modulo F/S

Cop 3. Trabajo

* Los reg. int. del PIC se sitúan a partir de 20h.

Accesos c/ op L/E. Int. hardware asignadas:

F01: pl. romandos escribir, pl. fin int. escribir 20h

IMR: máscara de interrupciones 1/3

IRR: pol. de int. Andro 1/3 en 1

ISR: int en serv.

INT0 INT1: du 1/3 ve.

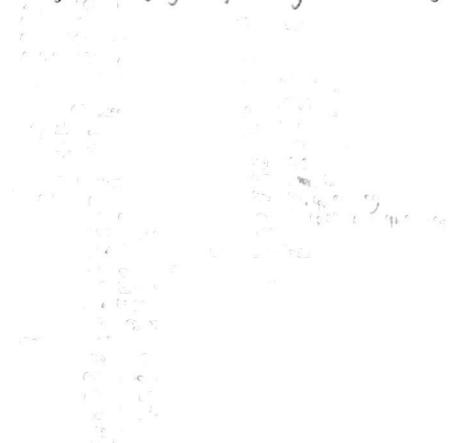
Práctica 2

Cuando un disp. quiere interrumpir debe enviar lo siguiente: señal/pedido interrupción y su identificador. Se usa el PIC, configurar interrupciones habilitadas y donde se maneja su rutina.

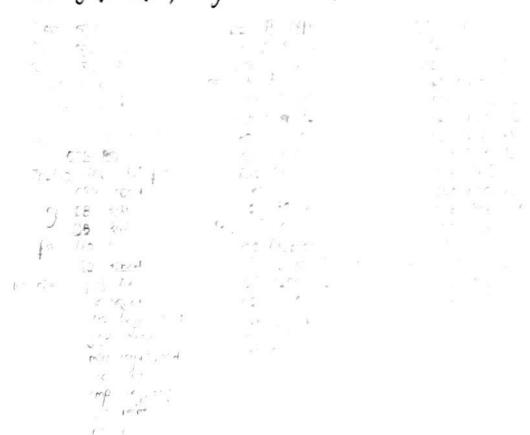
Una interrupción ocurre en cualquier momento, antes de ir a la rutina de la interrupción se configura en plato: flags y IP (dirección del rutina)

ac0ch	ac1ch	ac2ch	ac3ch	ac4ch	ac5ch	ac6ch	ac7ch	ac8ch	ac9ch	ac10ch	ac11ch	ac12ch	ac13ch	ac14ch	ac15ch	ac16ch	ac17ch	ac18ch	ac19ch	ac20ch	ac21ch	ac22ch	ac23ch	ac24ch	ac25ch	ac26ch	ac27ch	ac28ch	ac29ch	ac30ch	ac31ch	ac32ch	ac33ch	ac34ch	ac35ch	ac36ch	ac37ch	ac38ch	ac39ch	ac40ch	ac41ch	ac42ch	ac43ch	ac44ch	ac45ch	ac46ch	ac47ch	ac48ch	ac49ch	ac50ch	ac51ch	ac52ch	ac53ch	ac54ch	ac55ch	ac56ch	ac57ch	ac58ch	ac59ch	ac60ch	ac61ch	ac62ch	ac63ch	ac64ch	ac65ch	ac66ch	ac67ch	ac68ch	ac69ch	ac70ch	ac71ch	ac72ch	ac73ch	ac74ch	ac75ch	ac76ch	ac77ch	ac78ch	ac79ch	ac80ch	ac81ch	ac82ch	ac83ch	ac84ch	ac85ch	ac86ch	ac87ch	ac88ch	ac89ch	ac90ch	ac91ch	ac92ch	ac93ch	ac94ch	ac95ch	ac96ch	ac97ch	ac98ch	ac99ch	ac100ch	ac101ch	ac102ch	ac103ch	ac104ch	ac105ch	ac106ch	ac107ch	ac108ch	ac109ch	ac110ch	ac111ch	ac112ch	ac113ch	ac114ch	ac115ch	ac116ch	ac117ch	ac118ch	ac119ch	ac120ch	ac121ch	ac122ch	ac123ch	ac124ch	ac125ch	ac126ch	ac127ch	ac128ch	ac129ch	ac130ch	ac131ch	ac132ch	ac133ch	ac134ch	ac135ch	ac136ch	ac137ch	ac138ch	ac139ch	ac140ch	ac141ch	ac142ch	ac143ch	ac144ch	ac145ch	ac146ch	ac147ch	ac148ch	ac149ch	ac150ch	ac151ch	ac152ch	ac153ch	ac154ch	ac155ch	ac156ch	ac157ch	ac158ch	ac159ch	ac160ch	ac161ch	ac162ch	ac163ch	ac164ch	ac165ch	ac166ch	ac167ch	ac168ch	ac169ch	ac170ch	ac171ch	ac172ch	ac173ch	ac174ch	ac175ch	ac176ch	ac177ch	ac178ch	ac179ch	ac180ch	ac181ch	ac182ch	ac183ch	ac184ch	ac185ch	ac186ch	ac187ch	ac188ch	ac189ch	ac190ch	ac191ch	ac192ch	ac193ch	ac194ch	ac195ch	ac196ch	ac197ch	ac198ch	ac199ch	ac200ch	ac201ch	ac202ch	ac203ch	ac204ch	ac205ch	ac206ch	ac207ch	ac208ch	ac209ch	ac210ch	ac211ch	ac212ch	ac213ch	ac214ch	ac215ch	ac216ch	ac217ch	ac218ch	ac219ch	ac220ch	ac221ch	ac222ch	ac223ch	ac224ch	ac225ch	ac226ch	ac227ch	ac228ch	ac229ch	ac230ch	ac231ch	ac232ch	ac233ch	ac234ch	ac235ch	ac236ch	ac237ch	ac238ch	ac239ch	ac240ch	ac241ch	ac242ch	ac243ch	ac244ch	ac245ch	ac246ch	ac247ch	ac248ch	ac249ch	ac250ch	ac251ch	ac252ch	ac253ch	ac254ch	ac255ch	ac256ch	ac257ch	ac258ch	ac259ch	ac260ch	ac261ch	ac262ch	ac263ch	ac264ch	ac265ch	ac266ch	ac267ch	ac268ch	ac269ch	ac270ch	ac271ch	ac272ch	ac273ch	ac274ch	ac275ch	ac276ch	ac277ch	ac278ch	ac279ch	ac280ch	ac281ch	ac282ch	ac283ch	ac284ch	ac285ch	ac286ch	ac287ch	ac288ch	ac289ch	ac290ch	ac291ch	ac292ch	ac293ch	ac294ch	ac295ch	ac296ch	ac297ch	ac298ch	ac299ch	ac300ch	ac301ch	ac302ch	ac303ch	ac304ch	ac305ch	ac306ch	ac307ch	ac308ch	ac309ch	ac310ch	ac311ch	ac312ch	ac313ch	ac314ch	ac315ch	ac316ch	ac317ch	ac318ch	ac319ch	ac320ch	ac321ch	ac322ch	ac323ch	ac324ch	ac325ch	ac326ch	ac327ch	ac328ch	ac329ch	ac330ch	ac331ch	ac332ch	ac333ch	ac334ch	ac335ch	ac336ch	ac337ch	ac338ch	ac339ch	ac340ch	ac341ch	ac342ch	ac343ch	ac344ch	ac345ch	ac346ch	ac347ch	ac348ch	ac349ch	ac350ch	ac351ch	ac352ch	ac353ch	ac354ch	ac355ch	ac356ch	ac357ch	ac358ch	ac359ch	ac360ch	ac361ch	ac362ch	ac363ch	ac364ch	ac365ch	ac366ch	ac367ch	ac368ch	ac369ch	ac370ch	ac371ch	ac372ch	ac373ch	ac374ch	ac375ch	ac376ch	ac377ch	ac378ch	ac379ch	ac380ch	ac381ch	ac382ch	ac383ch	ac384ch	ac385ch	ac386ch	ac387ch	ac388ch	ac389ch	ac390ch	ac391ch	ac392ch	ac393ch	ac394ch	ac395ch	ac396ch	ac397ch	ac398ch	ac399ch	ac400ch	ac401ch	ac402ch	ac403ch	ac404ch	ac405ch	ac406ch	ac407ch	ac408ch	ac409ch	ac410ch	ac411ch	ac412ch	ac413ch	ac414ch	ac415ch	ac416ch	ac417ch	ac418ch	ac419ch	ac420ch	ac421ch	ac422ch	ac423ch	ac424ch	ac425ch	ac426ch	ac427ch	ac428ch	ac429ch	ac430ch	ac431ch	ac432ch	ac433ch	ac434ch	ac435ch	ac436ch	ac437ch	ac438ch	ac439ch	ac440ch	ac441ch	ac442ch	ac443ch	ac444ch	ac445ch	ac446ch	ac447ch	ac448ch	ac449ch	ac450ch	ac451ch	ac452ch	ac453ch	ac454ch	ac455ch	ac456ch	ac457ch	ac458ch	ac459ch	ac460ch	ac461ch	ac462ch	ac463ch	ac464ch	ac465ch	ac466ch	ac467ch	ac468ch	ac469ch	ac470ch	ac471ch	ac472ch	ac473ch	ac474ch	ac475ch	ac476ch	ac477ch	ac478ch	ac479ch	ac480ch	ac481ch	ac482ch	ac483ch	ac484ch	ac485ch	ac486ch	ac487ch	ac488ch	ac489ch	ac490ch	ac491ch	ac492ch	ac493ch	ac494ch	ac495ch	ac496ch	ac497ch	ac498ch	ac499ch	ac500ch	ac501ch	ac502ch	ac503ch	ac504ch	ac505ch	ac506ch	ac507ch	ac508ch	ac509ch	ac510ch	ac511ch	ac512ch	ac513ch	ac514ch	ac515ch	ac516ch	ac517ch	ac518ch	ac519ch	ac520ch	ac521ch	ac522ch	ac523ch	ac524ch	ac525ch	ac526ch	ac527ch	ac528ch	ac529ch	ac530ch	ac531ch	ac532ch	ac533ch	ac534ch	ac535ch	ac536ch	ac537ch	ac538ch	ac539ch	ac540ch	ac541ch	ac542ch	ac543ch	ac544ch	ac545ch	ac546ch	ac547ch	ac548ch	ac549ch	ac550ch	ac551ch	ac552ch	ac553ch	ac554ch	ac555ch	ac556ch	ac557ch	ac558ch	ac559ch	ac560ch	ac561ch	ac562ch	ac563ch	ac564ch	ac565ch	ac566ch	ac567ch	ac568ch	ac569ch	ac570ch	ac571ch	ac572ch	ac573ch	ac574ch	ac575ch	ac576ch	ac577ch	ac578ch	ac579ch	ac580ch	ac581ch	ac582ch	ac583ch	ac584ch	ac585ch	ac586ch	ac587ch	ac588ch	ac589ch	ac590ch	ac591ch	ac592ch	ac593ch	ac594ch	ac595ch	ac596ch	ac597ch	ac598ch	ac599ch	ac600ch	ac601ch	ac602ch	ac603ch	ac604ch	ac605ch	ac606ch	ac607ch	ac608ch	ac609ch	ac610ch	ac611ch	ac612ch	ac613ch	ac614ch	ac615ch	ac616ch	ac617ch	ac618ch	ac619ch	ac620ch	ac621ch	ac622ch	ac623ch	ac624ch	ac625ch	ac626ch	ac627ch	ac628ch	ac629ch	ac630ch	ac631ch	ac632ch	ac633ch	ac634ch	ac635ch	ac636ch	ac637ch	ac638ch	ac639ch	ac640ch	ac641ch	ac642ch	ac643ch	ac644ch	ac645ch	ac646ch	ac647ch	ac648ch	ac649ch	ac650ch	ac651ch	ac652ch	ac653ch	ac654ch	ac655ch	ac656ch	ac657ch	ac658ch	ac659ch	ac660ch	ac661ch	ac662ch	ac663ch	ac664ch	ac665ch	ac666ch	ac667ch	ac668ch	ac669ch	ac670ch	ac671ch	ac672ch	ac673ch	ac674ch	ac675ch	ac676ch	ac677ch	ac678ch	ac679ch	ac680ch	ac681ch	ac682ch	ac683ch	ac684ch	ac685ch	ac686ch	ac687ch	ac688ch	ac689ch	ac690ch	ac691ch	ac692ch	ac693ch	ac694ch	ac695ch	ac696ch	ac697ch	ac698ch	ac699ch	ac700ch	ac701ch	ac702ch	ac703ch	ac704ch	ac705ch	ac706ch	ac707ch	ac708ch	ac709ch	ac710ch	ac711ch	ac712ch	ac713ch	ac714ch	ac715ch	ac716ch	ac717ch	ac718ch	ac719ch	ac720ch	ac721ch	ac722ch	ac723ch	ac724ch	ac725ch	ac726ch	ac727ch	ac728ch	ac729ch	ac730ch	ac731ch	ac732ch	ac733ch	ac734ch	ac735ch	ac736ch	ac737ch	ac738ch	ac739ch	ac740ch	ac741ch	ac742ch	ac743ch	ac744ch	ac745ch	ac746ch	ac747ch	ac748ch	ac749ch	ac750ch	ac751ch	ac752ch	ac753ch	ac754ch	ac755ch	ac756ch	ac757ch	ac758ch	ac759ch	ac760ch	ac761ch	ac762ch	ac763ch	ac764ch	ac765ch	ac766ch	ac767ch	ac768ch	ac769ch	ac770ch	ac771ch	ac772ch	ac773ch	ac774ch	ac775ch	ac776ch	ac777ch	ac778ch	ac779ch	ac780ch	ac781ch	ac782ch	ac783ch	ac784ch	ac785ch	ac786ch	ac787ch	ac788ch	ac789ch	ac790ch	ac791ch	ac792ch	ac793ch	ac794ch	ac795ch	ac796ch	ac797ch	ac798ch	ac799ch	ac800ch	ac801ch	ac802ch	ac803ch	ac804ch	ac805ch	ac806ch	ac807ch	ac808ch	ac809ch	ac810ch	ac811ch	ac812ch	ac813ch	ac814ch	ac815ch	ac816ch	ac817ch	ac818ch	ac819ch	ac820ch	ac821ch	ac822ch	ac823ch	ac824ch	ac825ch	ac826ch	ac827ch	ac828ch	ac829ch	ac830ch	ac831ch	ac832ch	ac833ch	ac834ch	ac835ch	ac836ch	ac837ch	ac838ch	ac839ch	ac840ch	ac841ch	ac842ch	ac843ch	ac844ch	ac845ch	ac846ch	ac847ch	ac848ch	ac849ch	ac850ch	ac851ch	ac852ch	ac853ch	ac854ch	ac855ch	ac856ch	ac857ch	ac858ch	ac859ch	ac860ch	ac861ch	ac862ch	ac863ch	ac864ch	ac865ch	ac866ch	ac867ch	ac868ch	ac869ch	ac870ch	ac871ch	ac872ch	ac873ch	ac874ch	ac875ch	ac876ch	ac877ch	ac878ch	ac879ch	ac880ch	ac881ch	ac882ch	ac883ch	ac884ch	ac885ch	ac886ch	ac887ch	ac888ch	ac889ch	ac890ch	ac891ch	ac892ch	ac893ch	ac894ch	ac895ch	ac896ch	ac897ch	ac898ch	ac899ch	ac900ch	ac901ch	ac902ch	ac903ch	ac904ch	ac905ch	ac906ch	ac907ch	ac908ch	

12) Programa de relaj segmentos y mante su funcionamiento dado por el



13) Reloj q arranca y detiene su marcha al principio suministrando FIO y g finaliza conteo a los 30°



mod = En f de control de ram y cuando se da dada el control de memoria

log de control de memoria

Clase 3 Problemas de E/S. Existe una gran variedad de periféricos con varios métodos

(uso tránsit de dif. sol de D en > velocidad, y usar > temporizadores y formatos de D)

En módulo de E/S realiza una interfaz entre el procesador y la memoria (bus) y los periféricos.

Puedes manejar J y más periféricos.

Dispositivos externos: E/S básicos: monitor, mouse, teclado

-Almacenamiento disco duro, CD, DVD
-Impresión, impresora y scanner

* Estado: comprueba el dispositivo: monitor, acceso interfaz de red

Hasta el momento: controlador, controladora, interfaz, adaptador de D

Controlador de E/S: controlador de D desde y hacia el módulo E/S

1) P/saber qué hacer con las señales

2) Manejo de D

3) Transductor: pl/converter D, convertir señales eléctricas a algún tipo de D.

4) Se le da a memoria lo transductor,

* Hoy una conexión mediante puentes

Control de un periférico: se da q es el módulo E/S

Interfaz entre el periférico y el módulo E/S

7 señales de control: E, /D, /RD, /WR, /CS, /RD, /WR

5 señales de control: /RD, /WR, /CS, /RD, /WR

4: de E: Ready/not ready

3: Control: I/O, manejo de direcciones.

3: Transferencia: manejo de D

2: Buffer: adaptador (1,8 de bits)

1: Interfaz de memoria

medio E/S interfaz entre el módulo y periférico

Coprocessadores de en módulo S/E

Control prop del dispositivo o lo CPU: temporizadores, formateo, doble buffering y de datos

Capacidades de E/S: temporizadores, formateo, doble buffering y de datos

Controlar o no las f/s del dispositivo

Controlador de E/S o procesador de E/S (manejo de puentes, importante de lo que se pide).

Controlador de E/S o controlador de dispositivo (manejo primario).

Módulo E/S: interfaz

✓ PC tiene control del módulo → se puede controlar modulo E/S o no int.

modo de operación

Operaciones de E/S: requieren 1º Discretamiento: E/S programada en memoria - E/S asistido.
Transferencia de info. lenta o rápida
Gestión de la transf.: mecanismos de sincronización y control de los hilos de D.

1) Discretamiento de E/S

- E/S programada en memoria (memory-mapped): los disp E/S y memoria comparten un único espacio de direcciones, E/S se pone en la memoria de L/E. No hay órdenes específicas p/ E/S.

- E/S asistido: espacios de direc separados; necesidad de líneas especiales de E/S y de memoria. Órdenes específicas p/ E/S. Espera p/ q lo haga o díjalo.

Técnicas de gestión de E/S: - E/S programada al espacio de memoria (DMA)

E/S programada: es un intercambio de D p/ CPU y el módulo. Lo CPU tiene control sobre lo q. de E/S. Comprobación del E del disp. Envío de comando de L/E - hilo de D. Lo CPU espera q el módulo E/S termine lo q. de D. Lo CPU permanece en modo durante un período.

Transferencia rápida

Detalles: Lo CPU solicita lo q. de E/S al módulo.

- El módulo E/S realiza lo q. de E/S.

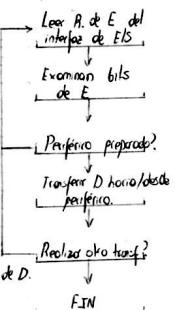
- El módulo E/S envía los bits de E del disp.

discretamente y, expuesto.

- Lo CPU comprueba periódicamente el E de estos bits,

hasta q lo q. de E/S sea realizado.

- En caso contrario, lo CPU espera y vuelve a comprobarlo + todo.



Órdenes de E/S: Lo CPU emite una dirección (específica el módulo).

Lo CPU da una orden:

Control: indica al módulo q hacer.

Test: comprueba el E del módulo y parámetros.

L/E: transfiere D desde o hacia el disp. x el bus de D.

E/S con interrupciones: lo CPU no tiene que esperar la finalización de la tarea de E/S, puede seguir procesando. No se repite la comprobación de los E. de los módulos. El módulo envía un pedido de interrupción a lo CPU cuando está lista para.

Qué hace lo CPU? Lo CPU envía una orden de lectura (El módulo E/S obtiene los D del periférico mientras lo CPU realiza otra tarea).

Lo CPU chequea si hay pedidos de interrupciones pendientes al final de q. alto de instrucción (FI). El módulo E/S envía un pedido de interrupción a lo CPU.

Lo CPU detiene el proceso, guarda el contexto, interrumpe el proceso y realiza la gestión de la interrupción.

Lo CPU solicita D (El módulo E/S transfiere los D).

Lo CPU recibe los D y sigue ejecutando su proceso.



bucle en libro

Cuestiones de diseño: Identificación del módulo q. interrumpe

• Disp líneas p/ d. módulo; PC, fondo no de disp.

• Consulta software (poli o exhaustivo): cuando un pedido de interrupc. lo CPU consulta a q. módulo p/ q. q. q. demandante. Es lento.

• Consiste en cadena ("daisy chain"/hard poll): lo línea de encadenamiento de interrupc. se conecta encadenando los módulos, la línea de pedido es compartida. Una vez enviado lo comienzo x parte de lo CPU, el módulo responderá (colorando un rc (polobito)), en el bus q. lo identif. La CPU amplía el rc como punto p/ acceder a lo rutina de servicio.

Interrupciones multíples: todos los líneas tienen un orden de prioridad. Los líneas q. de prioridad. Los líneas q. de prioridad pueden interrumpir a lo de < prioridad. Si hay un maestro del bus, sólo él puede interrumpir.

Estructura de interrupciones:

El Intel 8086 tiene solo 1 línea de petición de interrupc. y 1 solo controlador. Se deberá utilizar un arbitrio o gestor de interrupciones externo. Este chip tiene 8 líneas de interrupc., q. x lo cual podrá manejar 8 mod. E/S. Es útil el Interface de Periféricos programable (PIO). Es un módulo de E/S de propósito genérico. Tiene 24 líneas de E/S programable vía reg de control. Usado p/ una variedad de periféricos.

Análisis: las op. de E/S mediante interrupciones son + efectivas q. las programadas.

Ambas necesitan interrupc. del CPU.

Velocidad de transf. limitada en espacio + tiempo durante la operac.

Ej: E/S paralelo lento.

Pic. o 200 MHz (tiempo ciclo reloj = 5 ns); ciclos por instr. (PI=2), promedio.

Una instruc. tarda $\geq 5 \text{ ns} \times 2 \text{ instr} \Rightarrow 100 \text{ Mips}$

Quisiéramos imprimir un archivo de 10 Kbytes en una impresora láser de 20 pag x min.

1 pag = 3000 char (1 char = 1 byte). Imp. impresa 60000 char / min. 1 KB/s.

a) E/S q. espera de D, lo CPU entra en bucle y envía un nuevo byte q. vez q. la impresora esté preparada p/ recibirlo.

La impresora tarda 30 seg en imprimir 10 Kbytes.

(Nº copiado 30 seg) \Rightarrow podíamos haber hecho 3000 millones de instr.

b) E/S q. interrupciones, la impresora genera una interrupc. q. vez q. está preparada p/ recibir un nuevo byte.

Si lo gestionas bien, la CPU tiene 50 instr.

Al transferir 10 Kbytes tenemos q. enviar 80000 bytes.

30000 instr. \Rightarrow 1000 instr. \Rightarrow 1000000000 bytes.

Concluyendo, la E/S x interrupc. tarda en 10.000 veces el tiempo q. lo CPU se encarga de la impresora.

Alcuni criteri di valutazione

6699 d8 - 1007
113

Ej: E/S periférico rápido

Procesador a 200 MHz. Una instrucción 2-5 ms = 10 ns \Rightarrow ~500 Mips.
 Quedan transferir un archivo de memoria o disco de 10 Mbytes.
 El disco posee una velocidad de transf. de 10MB/s. (1 byte = 10⁻⁹ seg o sea nanoseg.)

- 1) El disco duro se encarga de recibir y enviar información al resto de componentes del ordenador.

b) E/S on interrupt

inst. $ATI = 10$ instruc. Pf kraf 300bytes, ejecuciones su² 10 ATI \Rightarrow ejecuts 300 millones de pf okrato al parafiso. \Rightarrow 1 seg.

Contur rápido: no mejoró tiempo.

Inciso de interrupciones.

Ambos tipos de interrupciones tienen en común que se identifican con un nro. de identificador de interrupción, y se requiere ejecutar una subrutina asociada a ese nro.

Hoy un VC de 255 dígitos me envió una observación de su interrupción y de su subrutina. Al recibir esa interrupción el identificador N correce o lo pasa. N de dicho VC, obtiene el dígito de comienzo de la subrutina y ejecuta un call a la misma. En efecto, el dígito N del VC se encuentra en N=4.

el plan ocupa 4 bytes (ns): primero almacenar la direc, los otros 2 no se usan

Intervenciones y resultados: INTD: Almacena el ID de intervox asociado al FIO

JNT 1: " fine
JNT 2: " Handshake.
JNT 3: " CIMA

Registro 44 es el único p/ instx in y out.

INT-1/func: 2 sectores: CONT y COMP, direc en coh y LH.

CONT se incrementa automáticamente una vez p/ registrar el tiempo transcurrido. CADA 200ms el tiempo límite del timer. Cuando CONT vuelve igual que se interrumpe.

(cont.) se vébo señicos: cuando se produce uno interrupt.

$\left\{ \begin{array}{l} \text{IMA: registro de configuración} \\ \text{ISA y IMA: registros de E.} \end{array} \right\}$

Practic₈ 3

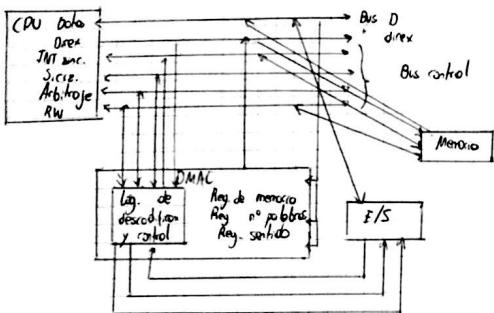
PJO interfaz de E/S que representa 2 puentes paralelos de 8 bits denominados A y B. Posee 4 reg. interno de 8 bits, PA y PB copian al D en los puentes A y B. "control; CA y CB manejan respectivamente los puertos A y B.

Corrección o partir de 304. $PA + PB = 1100$ (ant), $PB + CB = 1000$ (act)

Clase 3 part. 2

El controlador DMA (Acceso directo a memoria) es un dispositivo que controla una transferencia de datos entre el periférico y memoria sin intervención del CPU. Debe controlar tanto el bus como lo hace, y debe ser capaz de:

- Solicitar el uso del bus al serial y lograr su alta prioridad.
- Ejecutar la dirección de memoria sobre la que se realiza la transferencia.
- Generar señales de control del bus (modo general, señales de sincronización).



Etapas de una transacción DMA.

- Iniciación de la transferencia:
Lo CPU debe enviar al interfaz del periférico y al DMA los parámetros de la transferencia:
- Inicialización del interfaz (CPU-bus slave: interfaz): n bytes o transferencia completa (I/E).
 - Otro info (pistola, serial, etc.).
 - Inicialización controlador DMA ("CPU-bus slave: DMA): n bytes, tipo transferencia, dirección de memoria en modo pl. transf., n control (pl. DMA's y <> controladores).

Después de la inicialización, lo CPU libera sus líneas y ya no se ocupa de lo evolviendo de lo futuro.

Realizar de la transferencia:
Cuando el periférico está listo para realizar la transferencia, lo indica al DMA. Pide el control del bus y realiza la transferencia del periférico y la memoria.

Bus master: DMA + periférico. Bus slave: memoria.
Desp. de la transf. de pl. palabra, se actualizan las reg. del DMA: n bytes, direc mem.

Finalización de la transferencia:
El DMA libera el bus y devuelve el control al CPU. DMA solicita entonces una señal de interrupción pl. avisos al CPU la finalización de la op. E/S.

Problemas que puede haber:

Se pierde el rendimiento del CPU si el DMA hace uso intensivo del bus. Si el bus estuvo ocupado en una transferencia DMA, el CPU no puede acceder a memoria. El problema se reduce si el uso de memoria corta: la mayor parte del tiempo, la CPU hace instrucciones de lo contrario, y lo que no necesita usa el bus. El DMA puede aprovechar estos intervalos en los q' el CPU usa el control pl. para realizar las transferencias.

En caso de una IE sin cortes, el procesador no utiliza el bus en todas las fases de lo ejecución de una instrucción. El DMA aprovecha cuando el CPU no utiliza la memoria.

Tipo de transferencia:

Si el DMA sólo tiene el control del bus durante los intervalos de tiempo en los q' el CPU no hace uso del mismo, el rendimiento del sistema no sufre degradación alguna.

Hay 2 tipos de transferencia: refugio (burst), o por ciclo del ciclo (cycle-stealing).

Dos modo refugio:

El DMA solicita el control del bus a lo CPU cuando lo CPU concede el bus, el DMA no lo libera hasta finalizar la transferencia de todo el bloque de 0 completo. ventana de transferencia: durante el tiempo q' dura la transf. lo CPU no puede usar el bus de memoria y pierde el rendimiento del sistema.

DMA modo robo de ciclo:

Puede solicitar control del bus a lo CPU. Cuando lo consigue, se hace la transf. de una circa palabra y el DMA libera el bus. Puede solicitar bus tan vez q' sea necesario hasta finalizar la transferencia de bloques de D. ventana de transferencia: no hay degradación de rendimiento. desplazamiento de transferencia.

Pl. lo CPU no es una interrupción (pues no guarda contenido). Si bien el bloq es largo, no será tan largo como si ello realizara la transf. ∵ pl. transf. de E/S de múltiples palabras, es lo más eficiente.

Módulo controlador de periféricos:

Los dispositivos de E/S son cada vez más sofisticados. Al principio, CPU controla periféricos, se agrega un módulo E/S o controlador. a) Orden 2 + capacidad de interrupc. → El módulo E/S ejecuta DMA. Si módulo E/S tiene su propio procesador, c/s su pequeño conj. de instrucciones. b) El módulo ademas tiene memoria local, o sea se convierte en uno comp. en si mismo.

Control de controlador E/S:

- Representación: una extensión del concepto del DMA (menor habilidad de ejecutar instrucciones).
- Completo control de la transf. de D: lo CPU no ejecuta instrucciones E/S.
- Programa almacenado en memoria principal.
- Lo CPU inicia la transf. E/S (ordenar ejecutar el programa q' esté en memoria). El programa especifica dispositivos, áreas de memoria o usar, prioridades y acciones entre éstos.

1990 q contact City CB

Ejemplo - PIO.mp

PIO EQU 30H
 ORG 1000H
 HSJ DB "ORG. Y ARQ. DE COMP"
 FIN DB ?
 S B
 ORG 2000H
 mov AL,0FDH } CA = 1111 1101 → B = E S = S.
 OUT PIO+2, AL }
 MOV AL,0 } CB = 0000 0000 → PB = Solido
 OUT PIO+3, AL }
 IN AL,PIO } Fuerzo Strobe a O
 AND AL,0FDH }
 OUT PIO,AL }
 MOV BX, offset msj } Inicializo BX y C
 MOV AL,offset } fin. E. offset msj
 JNZ POLL { Leo. E. de BUSY y no avanza hasta q
 JNZ POLL { este libe (0)
 MOV AL,CBAN } Coloco char o imprimir en el
 OUT PIO+4, AL } PB
 IN AL,PIO } al lado de
 OR AL,02H } Fuerzo strobe a 1 genera
 OUT PIO,AL }
 MIN AL,PIO }
 AND AL,0FDH } Fuerzo strobe a 0
 OUT PIO,AL }
 INC BX } Apunto al sig char
 DCF DEL }
 JNZ POLL }
 INT0 }
 END
 6) ^{ORG 1000H}
 ;organizacion y arq de comandos
 fin
 ORG 2000H
 mov AL,07H
 OUT CA,AL
 mov AL,0H
 add AL,BAL
 add AL,offset fin - offset subetra
 mov AX,offset subetra
 END
 CND 1000H
 J2
 fin
 ORG 2000H
 OUT CA,AL
 JNZ POLL
 S DEC DEL
 add AL,03H
 OUT CA,AL
 DEC DEL
 JNZ POLL
 fin
 END

Impresario / P10

M_3d XXXXXXXSB R,E
 PB_1Q0 R,Q
 CA_321 XXXXXXX01 \rightarrow memory S,y,b
 CB_133 00000000
 Vwsgao .db: dBShtab bit S
 Busy bit E
 2) o Prasha 3
 PA F0U 32H
 PB F0U 31H
 CA F0U 32H
 ORG body
 Letto DB 'N'
 ORG 3000H
 strdec C: IN 'L',PA
 AND AL,0DHH \rightarrow JML 1000H
 OUT DL,AL
 RET
 ORG 3004H
 Leta D: IN 'P',PA
 OR AL,02H \rightarrow JML 1000H
 OUT DL,AL
 RET
 ORG 200CH
 mov AL,0DF4H \rightarrow JML 1000H
 OUT CL,AL
 mov AL,CH
 Out CB,AL
 D: in 'L',PA
 AND AL,I
 JNZ poll
 mov BX,CFH \rightarrow JML 1000H
 mov AL,T>I
 out BB,AL
 JML 1000H
 INT 0

d) ORG 500H
 CHAR DB ?
 BY 200CH
 MOV CL, S
 M
 MOV AL, 01H
 OUT CA, AL
 MOV AL, 04H
 OUT CB, AL
 RETUR CNP CL, S

```

d) for EQU 20H
    N_EQU EQU 40
    ORG 40H
    DB int_EQU
    ORG 100H
    dbc 20H
    ORG 200H
    Shi
    mov AL,01H
    out CA,AL
    mov AL,04
    out CB,AL
    subc CMP CL,5
    JZ fin
    poll: IN AL,PA
    AND AL,1
    JNZ poll
    call shobc
    mov BX, offset caec
    INT 6
    call shobc
    mov AL,100H
    out PB,AL
    call shobc
    DEC CL
    mov AL,0

```

3300H

```

    int_EQU DS:0000H AL, PA
    AND AL,1
    JNZ poll
    call shobc
    DEC CL
    mov BX, offset caec
    INT 6
    call shobc
    DEC CL
    JNE poll
    fin: MOV BX,0000H
    OUT EC0,AL

```

^{END}
close w: Sonnadox de instalações

Exercício 4. Argumentos de instituições
Sociedade civil e Estado: o que é o horizonte?

Sagittal de corte repto bival.

La segmentación de tráfico (pipelining) es una forma efectiva de organizar el hardware de la CPU para realizar + de un operación al mismo tiempo. Consiste en descomponer el procesamiento y ejecutar de las instrucciones en fases/etapas q permiten su ejecución simultánea. Si q se realizan varias instrucciones, cada etapa solo se usa de q.

Exploita el paralelismo q los instrucciones de un flujo secuencial pl. siguen o no siguen

Similar a la linea de ensamblado en una planta de tipo manufatura. El solo paso a través de él en el proceso de producir la vez que los productos pueden ser manipulados suavemente (figura E-2). Se puede comenzar el proceso nuevo (E o la linea de prod.) antes de que salga el prod. final de la misma.

- Lo segmentario es una forma de mejorar las prestaciones o niveles de desarrollo.
- Lo segmentario es visible en el programado.
- Necessidad de informar los plazos: el tiempo de lo + largo.
- El diseño de procedimientos segmentados tiene gran dependencia del requerimiento de resp.

Bien target Buffer provocado errores → se ejecuta una tabla, cuando sigue el bucle anterior

Biench Misoperación stolt (lleva a pend x memoria) SIS

Biench stolt como retardador → sólo entra de memoria no sirve si pide a pedir memoria RAM

B) A) Encara el loop porque no lo hace j/ciclos

B) se ejecuta el generador en memoria CPI = 22 ciclos, 12 instrucciones

C) dato word 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

RES: 0 A/D word 0x8000

cant: 0 word 0

code

ld 17, A/D[15]

ld 12, X[15]

ld 12, RES[15]

ib 16, RES[15]

je 17, 10, 10

loop: ld 17, R1[15]

and 12, 12

bnez 12, 10

jode 12, 10, 10

ld 12, 10, 10 RE

add 12, 12, 10

loop: add 12, 12, 10

mult 12, 12, 10

add 12, 12, 10

ld 12, 12, 10

ld 12, 12, 10

mult 12, 12, 10

add 12, 12, 10

halt

dato

word 4

on modo 4

on modo 0

on modo 2

on modo 10

on modo 12

on modo 14

on modo 16

on modo 18

on modo 20

on modo 22

on modo 24

on modo 26

on modo 28

on modo 30

on modo 32

on modo 34

on modo 36

on modo 38

on modo 40

on modo 42

on modo 44

on modo 46

on modo 48

on modo 50

on modo 52

on modo 54

on modo 56

on modo 58

on modo 60

on modo 62

on modo 64

on modo 66

on modo 68

on modo 70

on modo 72

on modo 74

on modo 76

on modo 78

on modo 80

on modo 82

on modo 84

on modo 86

on modo 88

on modo 90

on modo 92

on modo 94

on modo 96

on modo 98

on modo 100

on modo 102

on modo 104

on modo 106

on modo 108

on modo 110

on modo 112

on modo 114

on modo 116

on modo 118

on modo 120

on modo 122

on modo 124

on modo 126

on modo 128

on modo 130

on modo 132

on modo 134

on modo 136

on modo 138

on modo 140

on modo 142

on modo 144

on modo 146

on modo 148

on modo 150

on modo 152

on modo 154

on modo 156

on modo 158

on modo 160

on modo 162

on modo 164

on modo 166

on modo 168

on modo 170

on modo 172

on modo 174

on modo 176

on modo 178

on modo 180

on modo 182

on modo 184

on modo 186

on modo 188

on modo 190

on modo 192

on modo 194

on modo 196

on modo 198

on modo 200

on modo 202

on modo 204

on modo 206

on modo 208

on modo 210

on modo 212

on modo 214

on modo 216

on modo 218

on modo 220

on modo 222

on modo 224

on modo 226

on modo 228

on modo 230

on modo 232

on modo 234

on modo 236

on modo 238

on modo 240

on modo 242

on modo 244

on modo 246

on modo 248

on modo 250

on modo 252

on modo 254

on modo 256

on modo 258

on modo 260

on modo 262

on modo 264

on modo 266

on modo 268

on modo 270

on modo 272

on modo 274

on modo 276

on modo 278

on modo 280

on modo 282

on modo 284

on modo 286

on modo 288

on modo 290

on modo 292

on modo 294

on modo 296

on modo 298

on modo 300

on modo 302

on modo 304

on modo 306

on modo 308

on modo 310

on modo 312

on modo 314

on modo 316

on modo 318

on modo 320

on modo 322

on modo 324

on modo 326

on modo 328

on modo 330

on modo 332

on modo 334

on modo 336

on modo 338

on modo 340

on modo 342

on modo 344

on modo 346

on modo 348

on modo 350

on modo 352

on modo 354

on modo 356

on modo 358

on modo 360

on modo 362

on modo 364

on modo 366

on modo 368

on modo 370

on modo 372

on modo 374

on modo 376

on modo 378

on modo 380

on modo 382

on modo 384

on modo 386

on modo 388

on modo 390

on modo 392

on modo 394

on modo 396

on modo 398

on modo 400

on modo 402

on modo 404

on modo 406

on modo 408

on modo 410

on modo 412

on modo 414

on modo 416

on modo 418

on modo 420

on modo 422

on modo 424

on modo 426

on modo 428

on modo 430

on modo 432

on modo 434

on modo 436

on modo 438

on modo 440

on modo 442

on modo 444

on modo 446

on modo 448

on modo 450

on modo 452

on modo 454

on modo 456

on modo 458

on modo 460

on modo 462

on modo 464

on modo 466

on modo 468

on modo 470

on modo 472

on modo 474

on modo 476

on modo 478

on modo 480

on modo 482

on modo 484

on modo 486

1. Término para la parte que se ejecuta (excepto código y datos) } No depende de la función de ejecución
 2. Término que incluye el código y los datos } Se ejecuta en la memoria de los datos
- Técnicas dinámicas: recompilación salto/no salto: tablero basado en la historia de las instrucciones.
- Tabla de historia de saltos (Branch History Table o BHT): es una pequeña cache asociativa de la etapa de búsqueda (P) que tiene 3 campos: Dirección de destino del búfer, Info instrucción → que gobierna en la cache, y bits de E → se responde con cache.
- Salto retardado: realizar trabajo del mientas el salto se resuelve. Realizarlo en la memoria de datos.
- Aunque el trabajo de respuesta de salto (delay slot) es el periodo de respuesta o periodo luego de una instrucción de salto. El compilador toma en cuenta instrucciones que no dependen del salto en las etapas de respuesta. Si no es posible, se usan instrucciones NOP. Los datos en las etapas de respuesta de salto se capturan siempre.
- Requerir respuesta instrucciones.
- Otros subrutinas de hardware:
- Predictor si el código de ejecución: hay instrucción de salto. La tasa de acierto puede llegar al 95%.
 - Buffer de bytes (B):
 - Flujos múltiples (M):
 - Recopilador el destino del salto → prepara una lista de salto → sig. qd. Ejecutar lo que hace que se responda.
- Llamadas a subrutinas:
- 1) Muchas de entradas / salida. Muchas salidas.
- 2) Muchas múltiples: requiere copiar una x d'qd de salto), preparar el salto en > saltos. Se debe utilizar el manejo correcto.
- Asignaciones: para el acceso al bus y registros. Si hay múltiples salidas, se necesita un > nro. de salidas.
- Recopilador del destino del salto: se recopila la instrucción destino del salto, ordenemos de las instrucciones sig. o lo contrario. Las instrucciones se guardan hasta qd se ejecute la instrucción de bifurcación.
- 2) Buffer de bytes: memoria muy rápida. Gestiona qd lo largo de copias de instrucción del cache. Comprueba el buffer antes de hacer la copia de memoria. Usar qd para los datos que se van a copiar al pequeño buffer y salto. Usar qd para los datos qd se van a enviar.
- Cop 11 - Salto - Cap 1 Beltrán, Guzman
- Práctica 5 - explicar:
- Hasta ahora se utilizaban nro. de piloto fijo, lo cual incriminaba los CPI. Se encontró la solución de incluir nuevos componentes como un switch divisor y multiplexor. Permitiendo ejecutar instrucciones en el tiempo para seguirán nuevos problemas: stalls estructural / WAW / WAR.
- Solo se realizan 2 instrucciones en FX simultáneamente, tiene prioridad la 1^{ra} qd en la pipeline.
- WAW y WAR stalls: suceden cuando hay + de una unidad de ejecución dependiendo de D o 2 instrucciones qd entra al cache plantea sobrepujamiento entre instrucciones.
- El simulador genera estos errores cuando detecta algún potencial de conflicto.



(6.11) N (10: -)

Subrutinas + reducción gq x86. No hay manejo implícito de la pila (CALL y RET), no hay instrucciones explícitas pl Push y Pop.

Problemas:

- Registro de retorno: si el programa invoca a una subrutina y dentro de esa subrutina se invoca otra, el 2^{do} qd sobrescribirá el valor de qd de la nueva dirección de retorno.

- Sobrescribida de reg: si una subrutina usa registros de la subrutina qd la invocada, puede suceder qd los valores del mismo qd se conservan al regresar.

P/ resolver estos problemas, se establecerá una convención:

- Fx particular pl qd registro.
- Manejar qd qd subrutinas.

Registros + nombres y usos.

q0	siempre vale 0, no puede cambiarse	10
q1	Número de dirección - D.e. señales de salto. Debe ser salvado	11
q2	Vol de retorno de la subrutina llamada	12-13
q3	Argumentos pasados a la subrutina	14-15
q4-q7	Reg. temporales	(16-23) y (24-31)
q8	qd deben ser salvados	32-33
stqd	Stack pointer - Puntero topo de pila. Debe ser salvado.	34
frqd	Floating pointer - Puntero flotante del entorno. Debe ser salvado	35
fpqd	Pl. uso del kernel del SO	36
tpqd	Asamblea temporales - pl ensamblables	37-38
gpqd	Global pointer: puntero zona mem. estática. Debe ser salvado.	39

Ejemplos		Accesos de registro	
result, word0	.data	result, word0	Una subrutina debe garantizar qd preservar los val originales de qd
.text		.text	
daddi 10,10,10	→	daddi \$10,\$10,10	Así, una subrutina puede llamar otra qd subrutina salvando qd esto no modifica el val de reg.
decddi 15,10,20		decddi \$15,\$10,20	
jal simx		jal simx	Es necesario contar qd un lugar pl almacenar los val. originales de los reg.
sd 12, result(\$0)		sd \$12,result(\$10)	
halt		halt	
simx: daddi 12,14,15		simx: daddi \$12,\$14,15	se puede hacer en la mem. pl. qd una pila
j r3		j r3	
			Conviene usar de pila: qd no haber inst. específicas x convx todas las subrutinas usarán el qd. qd corre punto al topo de pila.
		push \$12 → daddi \$12,\$12,-8	
		pop \$12 → ld \$12,0(\$12)	

lo qd qd topo de pila.

Punto flotante tiene 32 registros de fd o f32.

Ej: calcular suma de val. del vc, multiplicar por 0.5 si es > 10, multiplicar por 2. Guardar en RS.

.data

vc: word 542,6

val0: double 0.5

val1: double 10

val2: double 2

rs: double 0

.code

add0: f1,0,0

add0: f2,10,4, /hola

f1,d {f1, res(f1), /hola

f1,d {f2, val0(f1), 0.5

loop:

ld (3, res(f1))

mul0: f3, f3, res(f1) * 0.5

add0: f3, f3, res(f1) + 0.5

mult0: f3, f3, res(f1) * 0.5

add0: f3, f3, res(f1) + 0.5

add0: f3, f3, res(f1)

add0: f3, f3, res(f1)

bnez f2, loop

// si es mayor a 10

f1,d f4, val0(f0)

c,f,d f3,f4, f4 = f4 de cada ciclo a f3

bctr f3,f4

f1,d f3, val0(f0)

mult0: f3,f3,f3

bne2 addi \$t2,\$t0,
 jr \$t0

9) date
 Lehrer, -er, -er
 vorlesen - lesen "lesen"
 result, -endo
 -ere
 Ich fah. fah. fah. [fah]
 ich e. e. e. e.
 Ich fu. fu. fu. fu. (fu.)
 fah.

e_vocal dadd 1v0,10,10
 loop loop 1v0,10,10
 e_vocal dadd 1v0,10,10
 loop loop 1v0,10,10
 e_vocal dadd 1v0,10,10
 loop loop 1v0,10,10
 e_vocal dadd 1v0,10,10
 loop loop 1v0,10,10

8) dodo: codro 1. codro 2. "codro"
radio: radio 1. "radio"
soft: soft 1. soft
cale:
dog: dog 1. dog, radio
doggy: dog 1. dog
el compo: el compo 1.
sd avo, heat 1. heat
holt:
remoppa: dodd 1. two
loop: loop 1. sug.
bu 1. \$1.
beg 1. el
bre 1. do. 1.
doddi: doddi 1.
dodd: dodd 1.
doddle: doddle 1.
doddli: doddli 1.

top:
fin - CO: fin 1. fin
dodd: dodd 1.
final: fin 1. fin

```

J0) .data
cadena: .asciiz "abc:ef"
resultado: .word 0
result: .word 0

.code
lbu $t0, cadena($0)
addi $t1, zero, 0x400
jal record
std $t0, result($200)

```

tenorido deddi st4, 10, f.
 kap: beginz st6, f/c
 l/w, deddi rodrat (10)
 deddi disp. disp. +
 sd 10, 0/1r
 fin l/w st5, 0/1sp)
 deddi, disp. disp. +
 deddi, 10, 10, st5
 bnez st4, loop
 1/10

$\delta\sigma_r \approx \sigma_r$

McGraw-Hill's *Principles of Economics* is the best book for AP® Microeconomics.

Close 6-11SC → ~~Objetos y cadenas de serie pedal batería para el organo~~
→ ~~Comunicadores de redacción fundador de las fucionarios~~

Historia de evolución
Homolo: > partes / = arquitectura pero / organizado
 > introducido en 1984 en System/300 en P4
 > siendo la arquitectura de los implementos

- Unidad de control microprogramada: idea de Wilkes en 1951.
- Memoria cache: en 1966.

RAM de estado sólido; que tiene menor espacio, más de sencillo manejo, y en los portátiles.

Microprocesadores: comienzo con Intel 4004 en 1971. Variantes: propósito gen. y entubado/empotrados.

- Procedural multiples: \rightarrow formal, often clear, sometimes more subtle

RISC computadoras de repertorio reducido de instrucciones.

Características propias: gran nro de registros de uso graf o mega terminología de computadoras pl optimizar el uso de los registros

- Apoyo de trabajo limitado y sencillos
- Enfasis en lo optimizar de la segmentación de informes
- Facilitar trabajo del escritor del compilador
- Mejorar la eficiencia y errores: secciones complejas de código intercambiable.
- Dar soporte a HLL y compiles

Iniciamente de 1982, el software es + corto que el hardware. Reduce peso y tamaño. Nv. de lenguaje / vez + complicado. Sólo sonido. dif. de ejecución HTML y op. de org.

Lado esto conduce a Repetición de instrucciones grandes + malas Varias sentencias de HLL implementadas en el hardware para la memoria

estudios dinámicos: medir durante la creación
 Operaciones: operaciones condicionales (IF, loop), el procedimiento llorando/rebano como frases,
 algunas instrucciones tienen un resultado que se usa en otra
 resultado: pudiendo escribir los resultados de las operaciones

Algoritmo de optimización de programas en el que se basa la compilación de los programas en una memoria.

Llamada y parámetros, se consume mucho tiempo, depende del nro. de parámetros.

También depende del nivel de anidamiento.

La mayoría de los program. no tienen larga secuencia de llamadas seguida x lo contrario.

La mayoría de los var. son locales, la ref. o operando están muy localizadas.

Conservarlos se pide ofrecer mejor separación pl. HLL optimizando las prestaciones de los correctos + usados pl. g + tiempo consumido. No es necesario q el algoritmo sea complicado. Usar gran nro. de reg./func.: optimizar función de operandos.

Pl. compilador: optimizar el diseño de las clases de instrucción (bitmask, predict, etc.). Recomendable un repertorio simple de instrucción const. de HLL o HLL+stack.

Algoritmo de optimización de programas en el que se basa la optimización de los registros.

Aplicación x software: el compilador es necesario pl. asignación de registros.

Asignación de registros a las variables q se usan + en un período de tiempo dado.

Requerir el uso de algoritmos de análisis de programas.

Aplicación x hardware: var. + registros + variables. Pueden intercambiarse en reg. durante períodos de tiempo + largos.

Registros pl. variables locales: muchos registros \Rightarrow reduce acceso a memoria.

Almacenar var. escalares locales en registros. \Rightarrow es + rápido.

Problema: cada llamada de proc./función cambia la totalidad. Deben ser pasados los parámetros, Atm. deben ser devueltos, las var. del programa deben ser devueltas.

Interfaz de registros: se requieren pocos parámetros y var. locales en pl. llamada.

Hay límite en la "profundidad" de llamadas. Se considera que es grande cuando

Usa múltiples conjuntos de registros. Pl. pl. llamada \Leftrightarrow La llamada no cambia el conj. de reg. o cosa.

Los registros suelen a cambiar el criterio conjunto de reg. utilizado.

Hay 3 tipos dentro de un conj. de reg.: 1: Reg. de parámetros.
2: Reg. de D. locales.
3: "temporales".

Los reg. temporales de un conjunto se solapan q reg. de parámetros del nro. + bajo ordenante. Posibilita q los param. se pasen sin q exista mif. de D.

Interfaz de reg. locales

1	2	3	j
1	2	3	j+1

S. conj. de parámetros

Algoritmo de optimización de memoria.

Algoritmo de optimización de memoria. Toma hasta 6 mimbres q quiso aplicar más allá, se ejecuta en la VRD y lo q se traslada a memoria. Terminada la 7ma. instrucción, q se traslada a memoria.

El compilador asigna posiciones de memoria a las variables, informante pl. var. glob. o loc. q se crean frecuentemente.

Incorporar el procesador en registro de reg. pl. var. globales: divide reg de la memoria en cuadros.

Algoritmo de optimización de memoria.

Bloq.

Todos los D escalares locales.

Var. individuales.

Var. globales q el compilador

Salvo tablas/referencias basadas en profundidad de anidamiento.

Var. D - variables.

Direcciónamiento de registro

Cuadro

D escalares recibidos usados.

Bloques de memoria.

Var. glob. y las cuadros q el compilador.

Simplificación basada en algoritmo de compilación.

Mujos bloques de D.

Direcciónamiento de memoria

Algoritmo de optimización de memoria en el compilador.

Suponemos q reg. no es el reg. El uso optimizado es responsabilidad del compilador.

Los program. HLL no tienen referencias explícitas a la reg.

Anidado en -

Cada cuadro del programa candidato se asigna a un registro simbólico o virtual. Asignar el nro. simbólico de reg. simbólico q es un nro. fijo de reg. reales.

Reg. simbólico q no se solapan pueden compartir el reg. real.

Si se registran los reg. reales, algunos de los var. se asignan en posic. de memoria.

Algoritmo de optimización del compilador. Instrucciones compuestas son distintas de operación. Lo optimizar es difícil. Es lento + velocidad.

Programas + pequeños? el program. ocupan memoria pero en dia lo msn. es barato.

El nro. de bloq. de memoria q se ocupan no tiene porque ser + pequeño q tener + instr.

+ instrucciones cortas q op + largo.

Programas + rápidos? program. q usan instr. + sencillas, op + complejas, menor control del microprograma. q grande, q el tiempo de ejecución de instrucción simple.

No mimbres q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

q instr. q se ocupan dia la memoria no es tan lento q q instrucción simple.

$\Delta \mu = 10^{-1}$ ft/min \rightarrow mag 0.0056

8100 furt a 8100 no hay barra diferenciadora. Diccionarios tienen sucesión de entradas.
Categorización: fijo y fijo -contexto: comprender función de programación y uso del grafo.
-contexto: revisar el soporte de lenguaje de programación y uso
gráfico de AB VLSI
en el caso de programación se usa la idea de procedimientos y subprocedimientos.

to form the first joint committee.

600 p.m. 20 Sept.

8. The legend for the 1880's is as follows:

La respuesta de los organismos son una función de la

Close ?

41 501 a 452

mixed foods

CV⁴ C : 100 + 922

Práctica 6.

Control y Data son direcciones de puentes mapeados en memoria y son variables globales.

Control: .word 32 OnZero Carga los datos en registros mediante lwo \$0, Data(\$zero)
 lwo \$1, control(\$zero)

Proceso:

- 1 control → interpreto el resto de DATA como un nro 76 sin signo y lo imprime
- 2 control → DATA = 72 / signo y se imprime
- 3 control → DATA pto. flotante e implemento
- 4 control + DATA son los dígitos de una cadena formada en 0, imprimo string
- 5 control → DATA resta 16KB (4 bytes), en DATA+4 la coordenada Y y en DATA+5 coordenada X, se pinta el pto indicado. Resolv 50x50.

- 6 control → limpia portillo alfanumérico
- 7 control → gráfica
- 8 control → permite ingresar más de un pf., y el valor se podrá leer en DATA.
- 9 control → se podrá pines encarar uno por uno en dato
- 10 control → el código ASCII de la letra teclea en DATA. Debo leer DATA con IBSU

se leerá

Puntero pixel

- 1) Puntero en DATA[0] DATA + 3 el resto RGBA(4 bytes)
- 2) DATA + 4 coordenada y
- 3) DATA + 5 coordenada x

BB
BB
DD

Winwips procesador tipo RISC. Instrucciones de 32 bits, registros de 64 bits. No hay flags de estado. Arquitectura hardware.

Tiene 32 reg de 32 bits para operaciones y 32 de PF. 16 son reg cualquier

Threading → incrementa productividad pero no reduce tiempo de ejecución

Podemos hacer op de E/S mediante el mpu

E/S se DATA y disp para valor en control

control, MIA son salidas de memoria / se leen y escriben con instrucciones usando bus sharing / un hilo en salto de linea

RGB → 3 colores. 24 bits → 16 millones colores. $2^{24} \rightarrow 255$ colores por pixel $2^8 \times 3 = 6$ mill.

↑ panello grafico

Clase 7

Sistema de memoria: los programadores desean acceder

a la memoria / método de memoria rápido. /
lo que se ve es lo jerárquico de memoria. Organizado en niveles q son abiertos en <>
lugar físicos dentro del sistema / memo & q se gestiona de manera independiente

Objetivo: lo refer. del sis. tiene q ser lo del nv. más rápido al
costo del nv. más lento.
y medida q más el tiempo del crv, cada nv inferior es +
grande, + lento y + barato q el nv superior o superior
en la jerarq.

Debe haber correspondencia de direcciones en los <> nv.
Propiedades a cumplir

Inclusión: los D almacenados en un nv. han de estar
almacenados en los nv. inferiores o el

Coherencia: las copias de la misma info. en los <>
niveles deben contener las mismas val. /

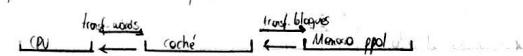
Per q se funcione la jerarq.: Si qd es cualquier una evolución de otra: coherencia q lo consta
caso de qd nv + bajo de jerarq. para q
Princípio de localidad de referencias: Tolerancia al nv + rápido

Localidad temporal: los elementos de memoria referenciados suelen volver a serlo en un futuro próximo. → subconjunto de nv.
En vez de todo un pabellón de nv, tiene un bloque
expansivo: los elem. de mem. cuyas direc estén próximas o las últimas referenciadas
serán referenciadas → subconjunto de nv



fabricado externo magnético y controlado → SD. / controlado
cache - MP - MV

fabricado y manejado separadamente y controlado del resto / ej: vaina pcie
Memoria cache: cantidad de memoria se ubica en la memoria ppal y CPU.
Puede alojarse en un chip o en módulo RAM.



funcionamiento de la cache: lo crv solicita contenido al d directo de memoria

Lo cache tiene q ser rápido. Sí, es así, lo obtiene de la cache (rápido).
Si no es así, se lo d. bloque q contiene esa direc desde la memoria ppal y copia
en la cache desp. lo crv entrega el d. requerido a la crv

Lo cache incluye etiquetas p/ identificar q bloq de la memoria ppal esté en el linea
concepto básico: Acerto (hit): se encuentra en la cache el d. solicitado.
Fallido (miss): no se encuentra en la cache el d. solicitado.

Un bloq q contiene la palabra. Accedido se copia de la memoria ppal a una linea
de cache.

• Tiempo q se tarda en un fallido depende de la latencia y costo de buso de la mem. ppal. O sea,
tiempo q determina q el bloq no esté en el cache y hallo el cacheado.

Latencia: tiempo p/ completar un acceso a memoria

Costo de buso: cost de info x unidad de tiempo q puede transferirse desde/hacia mem.

Fallas de cache: se gestionan mediante hardware y causan q el proceso sea detenido
hasta q el D esté disponible

Prestaciones de jerarq.: 7 accesos, 7 cache + 7 fallos memoria

7 fallos memoria = 700 dt fallos x penaliza dt fallo. $\Rightarrow 7 \text{ accesos} = 7 \text{ acceso} + 7F + PF$

P/ mejorar las prestaciones: Reducir el tiempo de acceso en caso de acerto

• Reducir el tiempo de fallo
• Reducir el costo de fallo

Supongamos q el dt de reloj 2.64Hz y CPI=1. gasto rdt de 300 ns.

Caso 1: hoy 2 mem. cache ideales M1 y M2 (no habrá fallo y tardará desparejo)

$$t = t_{CPU} + t_{mem} = t_{CPU} + 0 = t_{CPU}$$

$$t = nL \cdot CPI \cdot T = 100 \cdot 1 \cdot (1/26) = t = 50 \text{ ns.}$$

3 tipos de fallo de cache: vacío → los q no se referenciado cosa vacío → se produce q se refiere qd → cache no tiene qd. Los bloques en cache tienen
producción de conflictos → cuando bloques qd se localizan en cache y se viven
cambiando.

Forma de bloque: bloque grande por linea + folio

La solva es 2 niveles de cache
L1 > program el procesador, program y copiar
L2 > memoria Y + linea

Diseño de la cache: organiza (lomos y conj.), política de cifrar (tipo /x de correspondencia), política de remplazo (algoritmo de sustituir), política de escritura

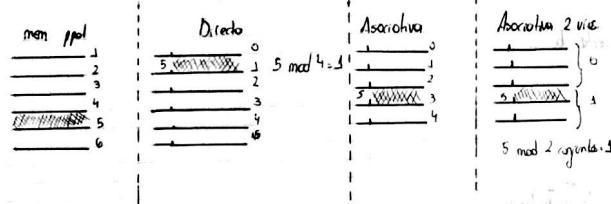
Organiza > lomos de cache y niveles. > lomos de cache si se requiere ofertar linea al folio
pac temporales grande pg el estor interbloque, memoria linea.

Ubicar de un bloque: y resto + grande = 1 linea

> correspondencia directa: un bloque solo puede estar almacenado en un lugar al cache.
Nº linea cache = nº bloque / nº mod = Nº líneas cache.

“ folio” asociativa: un bloque puede almacenarse en cualquier lugar de la cache

> asociativa x conjuntos: un bloque puede almacenarse en un conj. restringido de lugares en la cache. Un conjunto es un grupo de líneas de la cache.
Nº conj.: nº bloque / nº mod = Nº conj. cache



La interpretación de la dirección física depende del tipo que se use. Indire indirecto la linea o el conjunto q le corresponda. B0 representa todos los direc q no pertenezcan al bloque

Directo	Indirecto	Indice	B0
directa	indirecta	100	
Asociativa	directa	100	

* conjuntos directa indice B0

Correspondencia directa: simple, poco costoso. Hay una posibilidad de 1/1 bloques dado: si un programa accede a 2 bloques q se corresponden a la misma linea. (<> bloques de mem. ppd) de forma repetida. El perdido de cache será grande.

* asociativa: un bloque de MP puede colocarse en cualquier linea de cache. La etiqueta identifica unívoca un bloque de mem. Todo los direcc de la linea se examinan p/ buscar una coincidencia. Busqueda costosa

* asociativa de 2 vías: combina lo mejor de otros correspondencias. La cache se divide en un conj. de grupos. C/ conj. tiene un nro. de linea.

Un bloque almacenado corresponde a cualquier linea de un conj. determinado

1/1 directa q' existe d/ un folio

Política de remplazo: algoritmos de sustituir en correspondencia directa - en asociativa (LRU-FIFO)

Algoritmo de sustituir: de correspondencia directa. No hay der. Sólo hay una posible linea p/ 1/1 bloques. Se necesita una sustituir de esa linea (si o si).

Correspondencias coherentes: las algoritmos deben implementarse en hardware (p/ > ref.).

- LRU (menos reciente usado): requiere control de tiempo. Usa campo p/ saber cuándo viene usé una linea

- FIFO (primero en entrar primero en salir): requiere control de acceso se sustituye aquella linea q' ha estado + tiempo en la cache. Requiere algoritmo de FLS

- LFU (menos frecuentemente usado): requiere control de uso. Se sustituye aquella linea q' no se ha usado - referencias. Requiere control de tiempo

- aleatorio: se sustituye linea q' o azar.

Política de escritura: se debe evitar inconsistencias en el caso de escritura.

Tener en cuenta:

Lo q' se escribe sobre una linea de cache (el bloque de MP correspondiente debe ser actualizado en algún momento). Un módulo de I/O puede tener acceso directo a la MP.

En procesamiento paralelo: & múltiple CPU pueden tener caches individuales

Política de escritura: en directo.

Write-through (escribir en memoria inmediato): se actualizan simultáneamente la cache y la MP. Con múltiples CPU, observar el tráfico a memoria ppd p/ mantener actualizada el cache. Se genera mucho tráfico y retardo escritura

Write-back (post escritura): la info. solo se actualiza en cache. Se marca como actualizada > lat. de write. Lo q' MP se actualiza en el remplazo y puede contener info errónea en algún momento p/ que le haga escrituras innecesarias

Write-back o bypass de escritura p/ reducir latencia

Política de escritura: en folio:

Write allocate: lo q' se lleva de la MP a cache se subscrive en la cache. Habital q' ante la

No write allocate: el bloque no se lleva a la memoria cache. Se escribe directo a MP.

Habital q' write-through

Punto 4.- hace 3 nr. el corte. Corte L1 separado al D e instr.

o D dividido en 2 rutas reg. Corte x reg de 4 nr. Bloque de 64 byte. Frecuencia medida 1000

Lo corte de instruc. divide en segmentos al comienzo de grupo de ruta desencadenada

Corte L2, encadenado al D e I. 256 byte. Cdg. coordenado x e via. Bloques 512 byte. Algoritmo para enviar datos dentro de la ruta

Los 2 cortes en chip de procesador. Ancho de banda de L1 y L2 de 48 bytes/s.

los reg. permiten L3 en el mismo chip

procedimiento secuencial

stalling cap. 5. - Bettina cap 2 apuntado 2.1 o 2.4.

N y E

Use 6

Procesador escalon: ejecutar secuencial de instrucciones F, D, ALU, F, D, ALU, F, D, ALU

Segmentación de rutas: unidad segmentadora es un secuenc. de etapas al 1º la dependencia de que nuevos operaciones pueden iniciarse mientras otras estén en proceso

Procesador escalon segmentado: gráfica

Mayores prestaciones: lo evita la lógica de las divisiones de los procesos segmentados d/o liga o apunta de 2 términos de ejecución de > prestaciones

1) Procesador supersegmentado 2) Procesador superscalares. Cada nr. de instr.

Ejemplo supersegmentado: mucha q. no generan un ciclo de reloj, obtendremos > prestaciones subdividiendo el ciclo de reloj en sub-intervalos. p. 482

Resolución: una mayor frecuencia del ciclo de reloj; División de los etapas maestra del grupo segmentado en sub-etapas + pequeña (y más rápidas) y se transfiere D o lo > volatilidad del ritmo de reloj.

A tiempo / la instrucción individual no varía. 1º de grado de paralelismo, 2º de crecimiento paralelo

Ejemplo superscalares: se puede llevar a cabo f. de 4 instrucciones simultáneas. Cálculo de la duplicación de datos o todos los datos del PRU/ALU. Al captar múltiples instrucciones al mismo tiempo, gastos son y multiplicarán instrucciones ejecutadas (carga / alta tasa), mientras se lleva a cabo una q. alta.

El grado de paralelismo y crecimiento aumentan ya q. se ejecutan f. instrucciones en paralelo. Ello no obliga a que las q. de instrucciones sean iguales

Paralelismo de instrucciones: existe cuando instrucciones de uno secuenc. son independientes y pueden ejecutarse en paralelo solapándose. p. 483

Limitaciones: dependencia de D, dependencia relativa al procedimiento, conflicto de RR, dependencia de S, antidependencia. p. 483

Dependencias: se da si una instrucción depende de otra instrucción anterior

es creación dependencia, del d/corte y de reg. apuntar más en alto en el 2.4

Reg. = píxeles (o) en q. dirige la reg y la q. sigue, otro destino. Fase de manipulación de datos: se genera el orden de datos a su formación

Algoritmo de la secuencia: medida de la separación del procesador al sacar partida del trámite, interrumpir y localizar instrucciones independientes. Pista de instr. o $\frac{1}{2} \times 10^6$ s. Identificar paralelismo y agrupar F,D,E en paralelo. Remodelar el registro, romper la ruta. Sobre las instrucciones en la localización de instrucciones independientes: orden en que se reparten las instrucciones, cuán q. se ejecutan, cuán q. se actualizan las reg y posiciones de memoria

Uso óptimo del cache: ordenar independencias. Es importante orden de ejecución y

Algoritmo de emisión en desordenadas superscalares: emisión y finalización en orden, emisión en orden y finalización desordenada. Ejemplo gráfico

Reg. 1) MW (dependencia verdadera)

la salida de (1), R3, es un operando en (2)

2) WMR (dependencia de salida)

(1), (3) intentan escribir en R3

R3 > R3 op RS (1)

R4 > R3 op (2)

R3 > R3 op (3)

R7 > R3 op R4 (4)

3) WMR (antidependencia): instr. (3) no puede comenzar hasta q. la instr. (2) haya terminado uno de sus períodos (R3)

Flujo de reg. Los dependencias de salida y antidependencias surgen por la volatilidad de los registros. No pueden reflejar la secuencia de volatilidad dictada. x el flujo del programa. Esto puede detener alguna etapa del cache. Cuando la grilla de una instrucción guarda en el cache, se almacena en un reg. nuevo > remontamiento de reg. p. 483

La reg. se cargan automáticamente, o sea, las referencias posteriores son los registros nuevos que surgen el periodo q. reg. anterior y el remontamiento de reg.

Implementar: ejemplos: las estrategias de captar simultáneamente de múltiples instrucciones y determinar dependencias entre los registros y mecanismos pl. para cada etapa

Mejorar: q. iniciar o entrar en múltiples instrucciones en paralelo. Al pl. la grilla en paralelo de múltiples instrucciones. Mecanismos pl. para q. el procesador en un orden correcto

Consideraciones debatible en el proc. superscalares:

1) Influencia de la excepción: En el momento en q. se produce una excepción hay varios instrucciones en ejecución. Si q. se produce una excepción

2) Repetición: implica q. el comportamiento futuro sea idéntico al que tenía b. mero

computadora no segmentada.

P/ garantizar un estado consistente (proceso): instrucciones terminan escritas, lo q' comienza siguiendo la excepción y siguiente se abordan, tras lo ultimo de tratamiento se sigue lo q' siguió la excepción.

Ejecuciones paralelas: ej: las tareas se alternan en el ciclo en q' operan las instrucciones.

Retardo del W2 hace ciclo 6. Con las máximas interrupciones externas => exponencial paralelo.

La unidad de emisión sigue el orden y se recorre la tabla. Todas las instrucciones pendientes se completan => cambia el prior de interrupción.

Compromiso entre:
 - Ejecución desordenada => libera la unidad de ejecución.
 - Completar instrucciones en orden => excepciones paralelas.

Retardo sobre emisión desordenada y finalizar ordenada => se hace problema.
 Ejemplo de ejecución temporal (TW) => Cambio de orden.

El orden temporal (TW): se usa solo < inst posterior a i

Cop. 13: stalling; cop. 3: bucle. Recalculo -> q' las otras se ejecutan a la vez

Máx. comparsión memoria

Ciclo 9:

Anexo: bús del sistema: memoria, módulo de E/S, procesador.

• La memoria le envía D, recibe dirección y señales de control como reloj, o I/E, para enviarlos a la memoria.

• Módulo E/S: recibe y entrega D, interactúa c/ el procesador y memoria.

Recibe dirección y señales de control (reloj y genero ej. ATA)

• Procesador: L/E en memoria, genera/cambia señales,

• Bús de D: acto de bus s/cart. se alternan transferencia n D simultánea

• Bús de direcciones: placa D o instrucción que determina capacidad de transferencia

• Bús de control: señales de UC y otros componentes

• Un bús multiplexado tiene pl. bus de D, e instrucción. Requiere un horario de control y como resultado de eso, un bús pl. D e I, el cual será más chico debido a - circuito y un sust. + chico.

• Arbitraje: coordinar de ambos: pl. control de ambos -> controlador MMU.

El control del bús de D puede ser centralizado o distribuido.

Centralizado: una única unidad controla

Distribuido: cualquier parte nombrarse dentro y coordinar

Ventana 1 f. notacional => obturador de D

Ciclo 9: procesamiento paralelo. Arquitectura q' varios procesadores. p/ los q' siguió existiendo. Seo q' el nivel de prestaciones, la demanda de máquinas q' > rendimiento. Mejorar el rendimiento de una máquina q' en solo procesador => paralelismo o mult. instruc. Arquitectura de sist. q' varios procesadores => paralelismo o mult. proceso

Término q' q' q' de Tercer mundo. Categorías de computadoras:
 - SIMD: una secuencia de instrucciones, y 1 secuencia de D.
 - SIMD: uno secuenc. de I y múltiples secuencias de D.
 - MISD: múltiples secuencias de I y secuencia de D.
 - MMID: múltiples secuencias de I y múltiples de D.

SISD: un único procesador interpreta una única secuencia de I. Los D se alternan en una única memoria sin multiprocesador.

SIMD: un único instrucción q' máquina controla paso a paso la ejecución simultánea de un grupo nro de elem. de proceso. Si elem. tiene una memoria dedicada / q' I, es gestionada x el procesador, q' un conjunto de D => procesadores vectoriales y matriciales?

MISD: q' vectorial, o paralelo. Computadoras q' uno unico UC y una matriz de elem. "computacionales".

Tipo de instrucciones q' el procesador:
 - Ejecución de los instrucciones escalares: sucesivamente, multiplicador: se calculan n q' secuencias ejecutadas en todo los procesadores simultáneos.

Debe evitarse la necesidad de transferir el conj. de instrucciones D entre los procesadores y ventanas q' procesadores.

MMID: se trasmite una secuencia de D o un conj. de procesadores. El procesador gesticula una secuencia de instrucciones => Esto estruct. nunca se implementó.

MMID: conj. de procesadores gesticula secuencias de I => en simultáneo conj. de D => se puede dividir si tiene q' memoria:

- Mem. compartida: sup (multiprocesadores simétricos) y sist. NUMA
- Mem. distribuida: clusters.

Multi procesador simétrico: sup computadora caliente q' trabajo grande

- 2 o + procesadores similares q' ejecución comparables
- Comparten lo up y lo E/S
- Interconectados mediante un bús u otro tipo, qd. sist. de interconexión.
- Tiempo de acceso o memoria similar q' todo los procesadores.
- Todo los proces. pueden disponerse q' f.
- So intercambio q' información interna q' proc. y programas

o.8 ó "b" > p.9 o "c" en simultáneo

- Mejores prestaciones: si el bus o reloj se realizan puede organizarse en paralelo.
- Buena disponibilidad: un fallo en un proc. no detendrá lo comp.
- Incremento incremental: se pueden añadir + procesadores.
- Escalado: se fixa la cant. de procesadores.
- Cambio: bus compatible

Desarrollo: bus tiempo compartido: plazos limitados x el tiempo de ciclo del bus.

Reloj procesador obtiene este equipamiento de una memoria cache pl. mejorar las prestaciones: sofware el uso de acceso. Se pueden producir problemas de coherence de cache. Este problema debe ser resuelto x el hardware → protocolo de control y de directorio.

Clusters: computadoras completamente interconectadas q/ trabajan conjuntamente como un único AR. q/ computadora se denomina nodo. Prestaciones y disponibilidad elevadas. Aprox. igual de un servidor. Memoria o swap.

Beneficios: escalabilidad absoluta, incremental. Alto disponibilidad. Mejor relación precio/prest.

Cluster vs SMP ^{6.23}
Ambas: dan soporte a aplicaciones de 1 dominio de AR. Disponible coherentemente.
SMP: facil de administrar y configurar cache o sistema en solo procesador. Lo más fácil es lo dif. prop., menor espacio físico, > consumo de potencia.

Cluster: superior escalabilidad incremental y absoluta. Superior disponibilidad, redundancia.

Términos: CRASH, NUMA, CC-NUMA
toda los procs. tienen acceso a toda la mem.

• NUMA - Uniform memory access: igual tiempo de acceso a todos los registros de memoria, = tiempo de acceso a memoria q/ <> procesadores

• NUMA - Non-uniform memory access: el t. de acceso difiere > reg. de mem. accesible, <> procesadores accedidos o reg. de mem en <> velocidad.

• CC-NUMA + cache coherente NUMA: es un NUMA q/ mantener coherencia de cache q/ los cache de <> procesadores.

Motivos NUMA:

SMP tiene límite producto de nro. de procesadores.
En clusters el nodo tiene AR: coherence de cache mantenida x software.

NUMA reduce costos de SMP y brinda mayor escala.

Obj. NUMA: logra memoria transparente del syst. y permitir mts. de </> bus o syst. conexas internas.



Graex CC-NUMA: q/ proc. tiene cache L1 - L2. q/ nodo tiene MP y están conectados los nodos q/ un tipo de red.

q/ proc. ve un único espacio de direcciones de memoria.
Cada uno accesa a memoria → cache L1, L2, MP, MV.

Funcionamiento multihilos: q/ paralelismo de instrucciones: sin el aumento de complejidad y consumo de potencia de los segmentos q/ cache y swapables. La ejecución de instrucciones se divide en secuenciales + pequeñas llamadas hebras q/ pueden ejecutarse en paralelo. Amplia libertad de diseño multihilos.

Termino: hebra y proceso:

concepto de hebra en procesadores multihilos: peste no es el de solo multiprogramación.

Proceso: un programa corriendo en una comp.

Prop. de AR: espacio de direcciones virtuales q/ otro caso: espacio de un proceso.

Conector de proceso

Hebra: unidad de tipo de un proceso q/ puede organizar, incluye un contexto de proceso y resto de D q/ su propio espacio de memoria.

Intercomunicable: q/ procesador comunica q/ otro hebra.

Comunicar de hebras: cambio de control del procesador q/ hebra de = proceso. Usualmente se hace q/ los conector de proceso

Multihilos implícito y explícito

Multihilos explícitos: graex, concurrente de instrucciones q/ > hebras explícitas. Mucha instrucción de > hebras en lenguajes compilados o > graex paralelo q/ cache paralelo.

Multihilos implícitos: graex concurrente de varios hebras establecidas de un único programador, definidos estéticamente x el compilador o dinámicamente < el hardware.

Procedimientos multihilos PC <> p. / q/ hebras q/ poder ejecutarse concurrentemente.
Se trata q/ hebra separada, plazo de salida q/ memoria de q/

Aproximadamente q/ graex multihilos real

hebra q/ memoria?

Cop. 56 slotting - Cop. 5 gramo