Práctica Final Organización del Computador

IMPORTANTE: Para aprobar el final es necesario tener correctamente resuelto el 60% del mismo. Las respuestas que no estén justificadas o estén mal justificadas se considerarán erróneas. Por favor lea bien el enunciado.

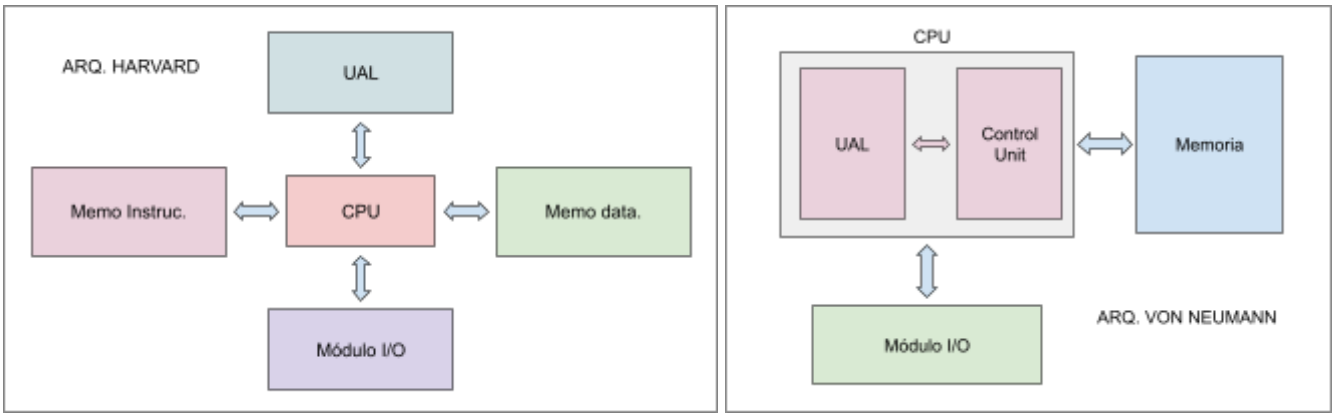
Las consignas son 7 por final y vale 2 puntos el que pide programar en ARM.

Final del 03/07/19

Harvard y Von Neumann

[1,5 ptos] ¿Cuáles son las ventajas y desventajas de la arquitectura Harvard en relación a la arquitectura Von Neumann? Grafique ambas arquitecturas y justifique.

La arquitectura Harvard propone romper con uno de los principios de la arquitectura Von Neumann, que es el principio de programa almacenado. Se plantea que existe un “cuello de botella” dado que tanto los datos como las instrucciones están almacenados en la memoria, por lo que para solucionarlo se separa y se almacenan por separado. Esto permitiria que sea más veloz al poder acceder a ambos en simultáneo. A su vez puede presentar una desventaja, porque si se quisiera modificar una instrucción en tiempo de ejecución. También al ser más complejo puede ser más costoso.



Ejecución condicional ARM

[1 pto] En la arquitectura ARM de 32 bits, ¿a qué se denomina ejecución condicional de una instrucción? De un ejemplo de su uso en assembler.

La arquitectura ARM se caracteriza por la ejecucion condicional de sus instrucciones, se basa en que segun los flags se ejecuta o no la instruccion. Esto es util para las bifurcaciones y para reducir los saltos.
Los 4 flags principales son: N (negative), Z (zero), C(carry) y O(overflow).

Las condiciones se especifican mediante un sufijo de 2 letras, por ej:
EQ, NE, GT, GE, LT…

Ejemplo en Assembler:
Se compara R0 y R1, en la instruccion de abajo se chequea que se cumple la instruccion MOV si R0 y R1 eran iguales (EQ), y la tercera linea hace el MOV R3,R4 si no son iguales (NE).

```
CMP     R0,R1
MOVEQ   R3,R2
```

```
MOVENE   R3,R4
```

ARM imprimir 3 cadenas

[2 ptos] Codificar un programa en assembler ARM de 32 bits que imprima tres cadenas de caracteres (definidas en el propio programa) por la salida estándar, haciendo uso de una subrutina interna.

```
    .equ SWI_PRINT_STRING 0x02
    .equ SWI_EXIT 0x11

    .data
string1:
    .asciz “Hola”
string2:
    .asciz “bello”
string3:
    .asciz “mundo”

    .text
    .global _start
_start:
    ldr        r3,=string1
    bl         SWI_PRINT_STRING
    ldr        r3,=string2
    bl         SWI_PRINT_STRING
    ldr        r3,=string3
    bl         SWI_PRINT_STRING
    b          fin
    @.end

print_r3:
    stmfd      sp!, {r0,lr}
    mov        r0, r3
    swi        SWI_Print_String
    ldmfd      sp!, {r0,pc}
fin:
    swi        SWI_EXIT
    .end
```

Formato variable Intel

[1 pto] ¿Por qué se dice que los formatos de instrucción de la arquitectura Intel x86 son variables? De algún ejemplo que justifique su respuesta.

Se dice que las instrucciones de la arquitectura Intel x86 son variables porque pueden tener longitudes y complejidades distintas. Cada instrucción está compuesta por una serie de campos de x cantidad de bits, en otras arquitecturas como ARM la suma de la cantidad de bits es siempre constante, por el contrario en intel hay muchos de esos “campos” que puede o no estar, por lo tanto la suma de bits puede dar distinto.
También el caso de las complejidades se puede ver en instrucción MOV
MOV ax,bx mueve entre registros y es “sencillo”

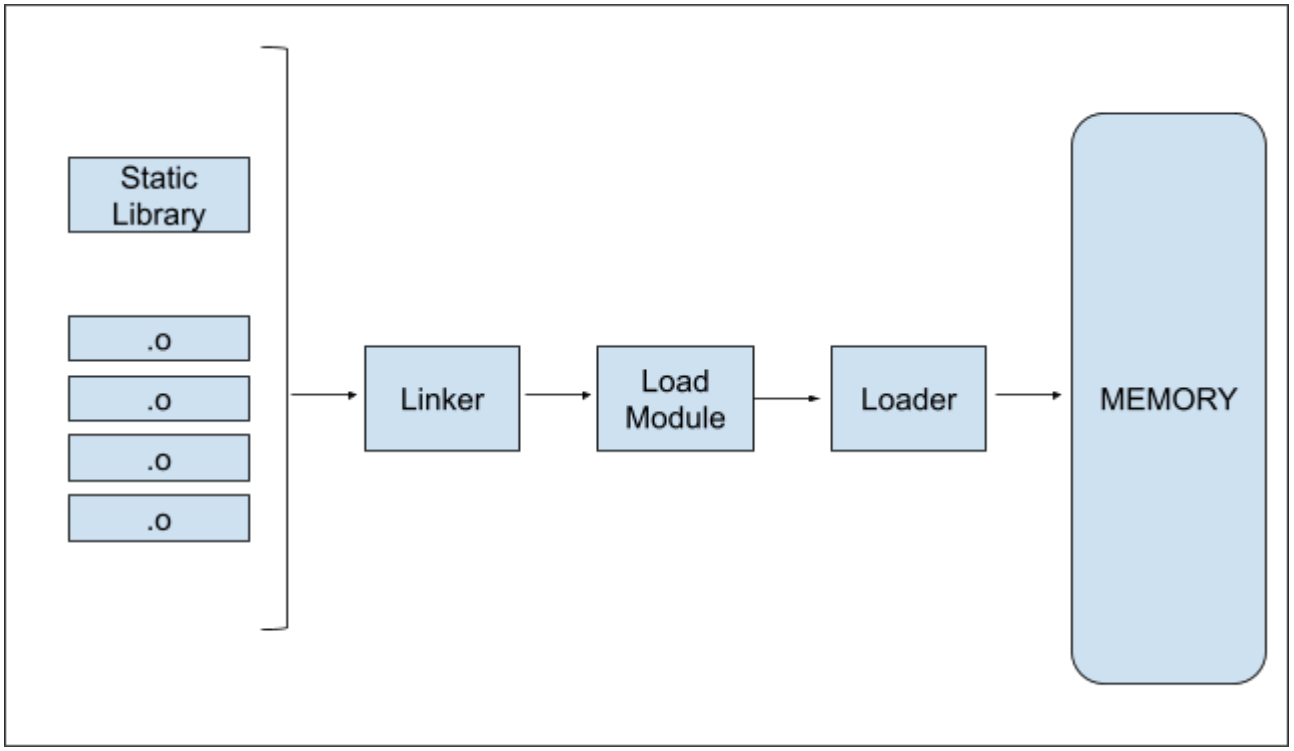
MOV ax,[1000] mueve al registro el contenido de la direccion mil, ya es un instruccion mas compleja.

MOV [bx],ax se mueve el contenido de ax, a la direccion de memoria almacenada en bx

Linking Estático

[1,5 ptos] Explique claramente cómo funciona el linking estático. Ejemplifique y grafique dicho funcionamiento.

El linking estatico es una forma en la que puede operar un linker. Los linker son los encargados de pasar todos los módulos de código objeto para generar finalmente el ejecutable. En el tipo estatico, se combinan por unica vez todos los módulos más el módulo de las bibliotecas necesarias, generando así un módulo de carga. Para luego pasar al “loader” donde se pasa a memoria y se inicia el proceso. Este tipo de linker genera un ejecutable pesado pero rapido, a diferencia de los dinamicos que generan un ejecutable mas ligero pero menos rapido ya que el ejecutable generado no tiene todas las librarys ahi, sino que tiene que ir a buscar las referencias en tiempo de ejecucion.



Ciclo de instrucción en procesador

[1,5 ptos] Explique claramente qué es un ciclo de instrucción en un procesador, indique qué etapas contempla y que ocurre durante la ejecución de cada una de ellas.

Un ciclo de instruccion de un procesador es el conjunto de etapas que pasa un procesador para ejectuar una instruccion de programa.

Las etapas son:

- Busqueda de la instruccion en la memoria, o registro similar que almacena la proxima instruccion a ejecutarse.
- Decode, el procesador decodifica que operacion debe realizar y que operandos necesita.
- Execute, donde se realiza la operacion indicada por la instruccion, puede ser aritmetica, logica, transferencia de datos, control de flujo, etc.
- Acceso a memoria, no es parte de todos los ciclos, pero es por si se desea almacenar por ejemplo un dato en memoria.
- Write Back, los resultados se reescriben en los registros.

Almacenamiento en cinta

[1,5 ptos] Identifique y explique cuáles son las principales desventajas del medio de almacenamiento en cinta. ¿Cuáles son sus aplicaciones actuales? ¿Qué ventaja comparativa tiene con respecto al resto de los medios de almacenamiento secundario?

Las principales desventajas de los almacenamientos en cinta son:

- Almacenamiento secuencial: para acceder a ciertas direcciones de memoria es necesario primero pasar por las que están antes
- Son lentas

Actualmente se utilizan como almacenamiento back up para grandes volúmenes de información a la que se accede solamente circunstancialmente.

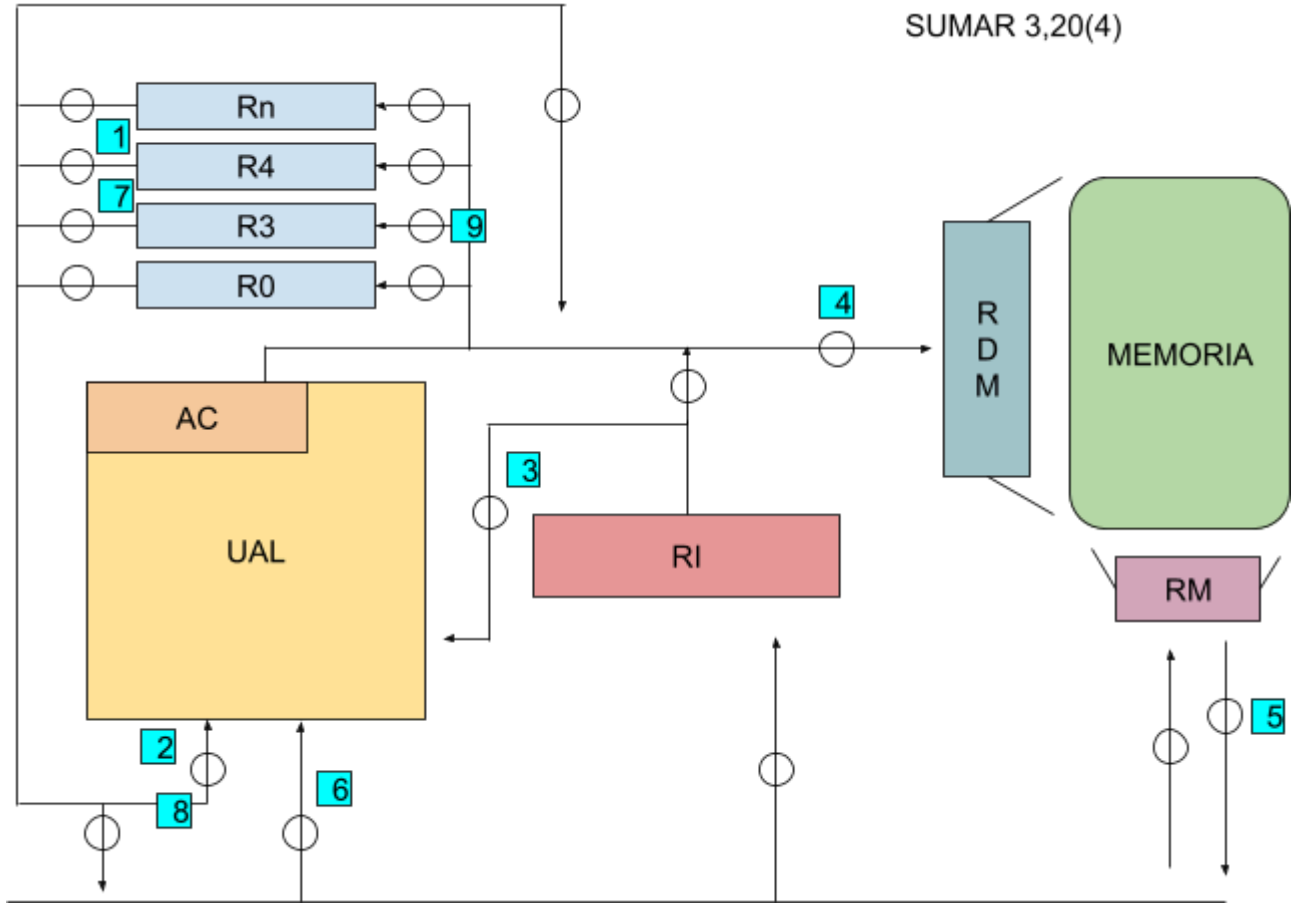
Las ventajas que presenta son:

- Gran espacio de almacenamiento.
- Vida útil, pueden llegar a durar 30 años
- Barato, no lleva elementos electrónicos lo que lo lleva a ser económico a comparación de los otros medios de almacenamiento.

Final del 09/09/20

SuperAbacus sumar 3,20(4)

[1,5 ptos] Indique cuales son las microinstrucciones necesarias para ejecutar la instrucción SUMAR 3,20(4) en la máquina SuperAbacus, siendo 3 y 4 registros de uso general y 20 un offset en base 10. Se pide además graficar en el esquema de la máquina el flujo de apertura de compuertas usadas en la fase de ejecución de dicha instrucción.



(R4) -> AC
(RI) -> AC
(R4) + 20(10) -> AC
(AC) -> RDM
((RDM)) -> RM
(RM) -> AC
(R3) -> AC
(R3) + (R3) -> AC
(AC) -> R3

Modos de direc. en ARM

[1 pto] Explique claramente y ejemplifique al menos cuatro modos de direccionamiento presentes en la arquitectura ARM 32 bits.

- Inmediato: mov r0,#4

Se carga el valor 4 de forma inmediata en el registro r0.

- Registro-Directo: mov r0,r3

La mov mueve (una copia) la data al r0 del r3.

- Directo: ldr r3,mem

La instruccion load register carga en el registro 3 un dato de memoria.

- Registro Indirecto: ldr r3,[r4]

Se carga el r3 con el contenido de la dirección contenido en el r4.

- Pre-indexado: ldr r3,[r4, #4]!

Se carga el r3 con el contenido de la direccion contenido en el r4 + 4(10).

- Post-indexado ldr r3, [r4], #4

Se carga el r3 con el contenido de la direccion contenido en el r4 y luego en r4 se incrementa en 4.

ARM imprimir 3 cadenas

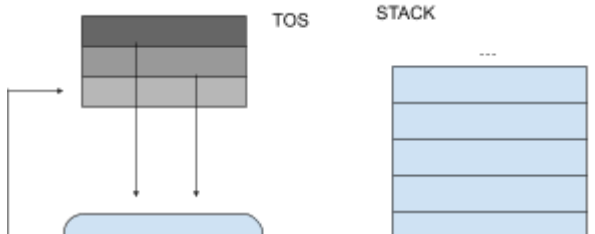
[2 ptos] Codificar un programa en assembler ARM de 32 bits que imprima tres cadenas de caracteres (definidas en el propio programa) por la salida estándar, haciendo uso de una subrutina interna. [Repetido]

Clasificación del repertorio de instrucciones

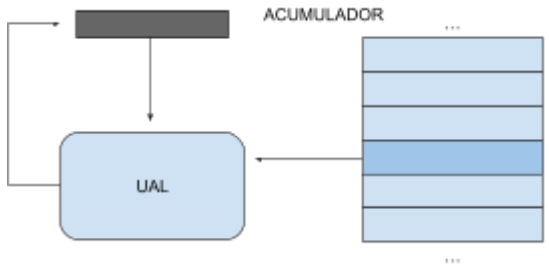
[1 pto] Indique como se puede clasificar el repertorio de instrucciones de una arquitectura de computadores de acuerdo a la ubicación de los operandos. Ejemplifique y/o grafique cada una.

El repertorio de instrucciones de una arquitectura se puede clasificar de acuerdo a la ubicación de los operando en:

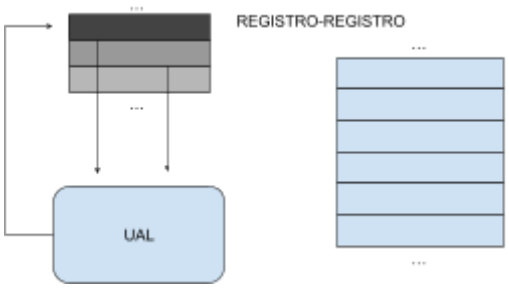
- STACK: (Intel, perdura por retrocompatibilidad)
  ej: PUSH A
  PUSH B
  ADD
  POP C



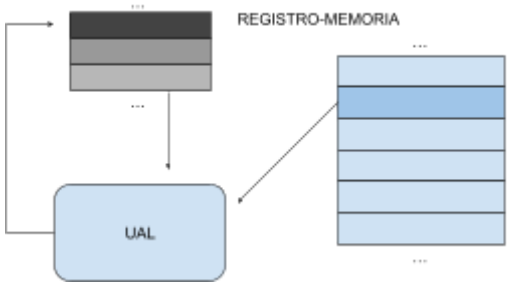
- ACUMULADOR: (Abacus)
  ej: LOAD A
  ADD B
  STORE C



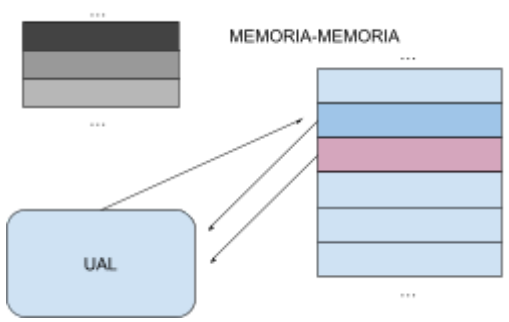
- REGISTRO/REGISTRO: (Intel)
  ej: LOAD R1,A
  LOAD R2,B
  ADD R3,R1,R2
  STORE R3,C



- REGISTRO/MEMORIA:
  ej: LOAD R1,A
  ADD R3,R1,B
  STORE R3,C



- MEMORIA/MEMORIA: (Intel, que sigue vigente por retrocompatibilidad, ARM no tiene)
  ej: MOVE C,A
  ADD C,B



Linking estático

[1,5 ptos] Explique claramente cómo funciona el linking estático. Ejemplifique y grafique dicho funcionamiento. [Repetido]

Procesamiento en paralelo por pipelining

[1,5 ptos] Explique claramente de que se trata la técnica de procesamiento en paralelo por pipelining en un procesador. Que complejidades pueden presentarse en el manejo de instrucciones y quienes pueden resolverlas. Grafique un pipeline de 5 stages.

La tecnica de procesamiento en paralelo por pipelining llego como solucion al tiempo que se perdia en cada instruccion en el procesador. Ya que antes de esta, cada instruccion debia pasar por las etapas del ciclo de instruccion del procesador. Lo que se propone es que cada etapa del ciclo se divide en módulos que se dedican a hacer esas tareas y entonces cada módulo puede agarrar varias instrucciones a la vez.

Fetch	1	2	3	4	5
Decode		1	2	3	4
Execute			1	2	3
Acceso a memo				1	2
Write Back					1

RAID de nivel 3

[1,5 ptos] En la arquitectura de discos RAID de nivel 3: ¿qué ocurre con las peticiones de E/S mientras un disco queda inutilizable? Explique además el algoritmo que permite recuperar la información perdida.

Cuando un disco se rompe, o queda inutilizable, en la arquitectura de discos RAID de nivel 3, lo que se hace a la hora de por ejemplo cargar nueva informacion en memoria, se hace de cuenta como si disco siguiera ahi en el calculo de paridad. De igual manera si se desea acceder a un dato que el disco que falta deberia tener.

Entonces, suponiendo que tengo 4 discos de almacenamiento y un quinto que exige la arquitectura. El calculo de paridad para sacar la serie de bits que tiene que almacenar ese quinto disco se calcula de la siguiente manera

$X4(i) = X0(i) + X1(i) + X2(i) + X3(i)$

Si llegara a faltar por ejemplo el disco 1. Para almacenar la informacion en el resto de los discos se hace de cuenta que si almacena lo que debe y se continua. Si quiero acceder a algo que deberia de haber almacenado es:

$X1(i) = X4(i) + X0(i) + X2(i) + X3(i)$

Y con ese mismo calculo cuando se inserte el disco que reemplace el que no funciona, se cargaran sus datos como si nunca se hubiese ido.

Final del 10/03/21

SuperAbacus sumar 3,20(4)

[1,5 ptos] Indique cuales son las microinstrucciones necesarias para ejecutar la instrucción SUMAR 3,20(4) en la máquina SuperAbacus, siendo 3 y 4 registros de uso general y 20 un offset en base 10. Se pide además graficar en el esquema de la máquina el flujo de apertura de compuertas usadas en la fase de ejecución de dicha instrucción. [Repetido]

Barrel Shifter

[1 pto] Explique claramente qué es y cómo funciona el barrel shifter en la arquitectura ARM de 32 bits. De ejemplos concretos con instrucciones assembler.

El barrel shifter es un componente de la arquitectura ARM que permite realizar operaciones de desplazamiento de bit de manera eficiente y rápida. Pueden ser tanto a izquierda como a derecha dependiendo de la dirección y cantidad de movimientos que se le pida, logrando efectos como multiplicar y dividir por potencias de 2.

Logical Shift Left

```
MOV R0, R1, LSL #1
```

Logical Shift Right

```
MOV R0, R1, LSR #1
```

Arithmetic Shift Right

```
MOV R0, R1, ASR #1
```

ARM vector “PAR”

[2 ptos] Codificar un programa en assembler ARM de 32 bits que recorra un vector de enteros y los imprima por la salida estándar agregando la leyenda “PAR” a continuación de todos aquellos números que así lo sean.

```
.equ SWI_Print_Int, 0x6B
.equ SWI_Exit, 0x11
.equ SWI_Print_Str, 0x69
.equ Stdout, 1
```

```
    .data
array_origen:
    .word 1,2,5,6
array_length:
    .word 4
eol:
    .asciz "\n"
par:
    .asciz " PAR"
    .text
    .global _start
_start:
    ldr r0, =array_origen
    ldr r2, =array_length
    ldr r2, [r2]
loop:
    ldr r4, [r0]
    bl imprimir
    add r0, r0, #4
    subs r2, r2, #1
    cmp r2, #0
    bne loop
    b exit
imprimir:
    stmfd sp!, {r0,r1,lr}
    ldr r0, =Stdout
    mov r1, r4
    swi SWI_Print_Int
    and r5,r4,#1    @Hago and en el último bit
    cmp r5, #0      @comparo para saber si es par o no
    bne impSig
    ldr r1, =par
    swi SWI_Print_Str
impSig:
    ldr r1, =eol
    swi SWI_Print_Str
    ldmfd sp!, {r0,r1,pc}
exit:
    swi SWI_Exit
    .end
```

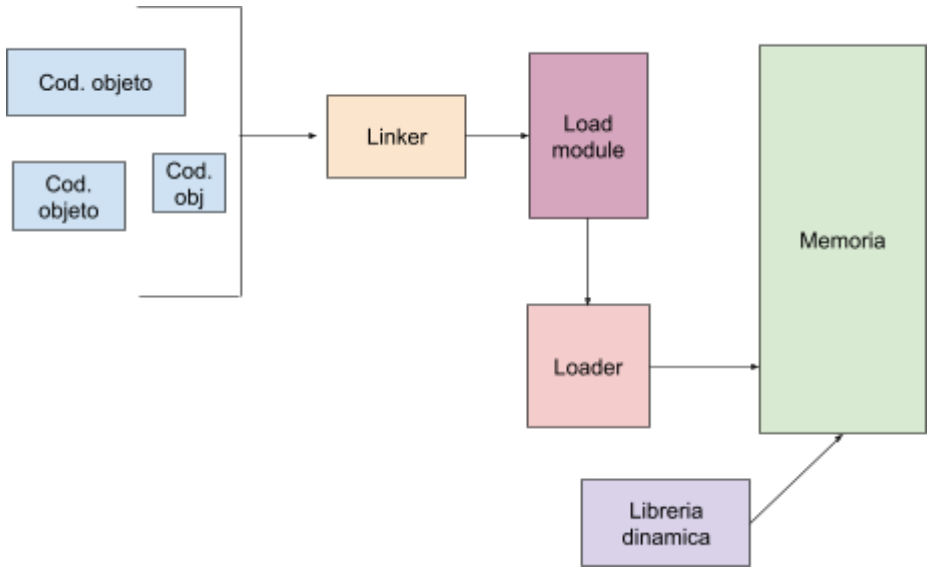
Formato variable Intel

[1 pto] ¿Por qué se dice que los formatos de instrucción de la arquitectura Intel x86 son variables? De algún ejemplo que justifique su respuesta. [Repetido]

Linking Dinámico en ejecución

[1,5 ptos] Indique claramente que es el linking dinámico en tiempo de ejecución y cuáles son las diferencias frente al linking estático.

El linking dinámico es un método de enlace que consiste en integrar las referencias externas del código principal al momento de ejecutar el programa en memoria. Es decir, cuando se realiza una llamada a una biblioteca externa, el sistema operativo busca y vincula dinámicamente el módulo llamador durante el tiempo de ejecución. A diferencia del linking estático, que compila los códigos fuente en código objeto y almacena estáticamente los archivos a los que hacen referencia, el linker prepara el módulo de carga con todo cargado, generando un archivo de ejecución más pesado. Sin embargo, al tener las referencias en el módulo, el proceso de ejecución suele ser más rápido.



Procesamiento en paralelo por pipeling

[1,5 ptos] Explique claramente de que se trata la técnica de procesamiento en paralelo por pipelining en un procesador. Que complejidades pueden presentarse en el manejo de instrucciones y quienes pueden resolverlas. Grafique un pipeline de 5 stages. [Repetido]

RAID nivel 3

[1,5 ptos] En la arquitectura de discos RAID de nivel 3: ¿qué ocurre con las peticiones de E/S mientras un disco queda inutilizable? Explique además el algoritmo que permite recuperar la información perdida. [Repetido]

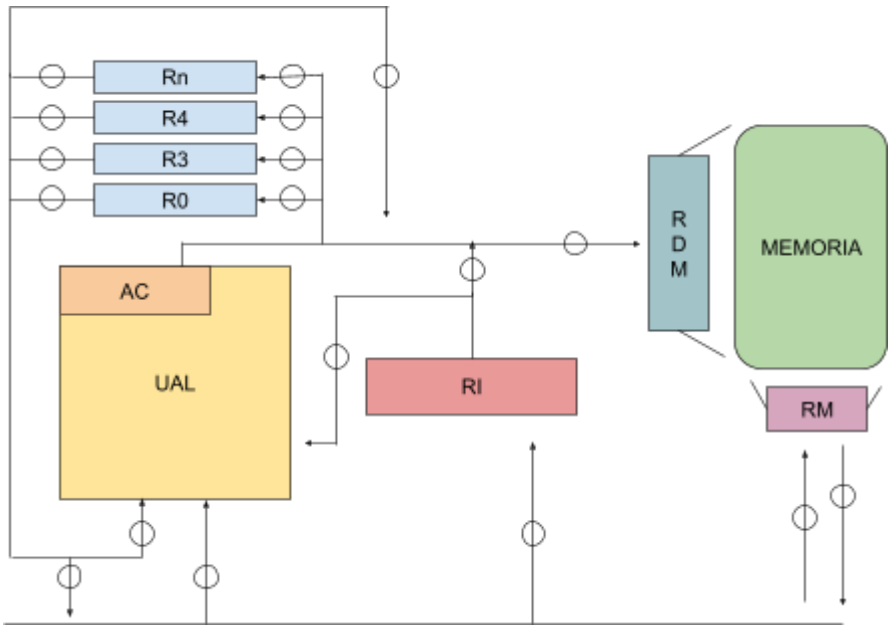
Final del 03/08/21

UAL en SuperAbacus

[1,5 ptos] Explique claramente por qué se dice que la UAL en Super Abacus se utiliza tanto para la suma de datos como de direcciones. Justifíquelo con microinstrucciones y el gráfico de la máquina.

Se dice que la UAL en super abacus se utiliza para la suma de direcciones también porque a diferencia de ABACUS, en esta arquitectura se puede almacenar en los registros direcciones. Es por ello que se permite instrucciones del tipo:
ADD [R3],20 donde se adicionaron 20 a la dirección almacenada en R3. Esto es útil para el movimiento entre elementos almacenados de forma secuencial.

En el gráfico de la maquina se puede ver teniendo acceso los registros a la UAL no hay nada que les impide participar en operaciones lógicas el contenido de las mismas.



Direccionamiento post-indexado

[1 pto] Explique claramente que es y cómo funciona el modo de direccionamiento post-indexado autoindexado (registro indirecto con post-incremento) en la arquitectura ARM de 32 bits. De un ejemplo concreto con una instrucción.

El modo de direccionamiento post-indexado es una forma de acceder a datos de forma eficiente. Se utiliza cuando se tiene un registro base con la que hacer una transferencia de datos y luego se desea desplazar la dirección a la que apunta el registro base.

Por ejemplo: LDR R3,[R4],#4

En este caso, el offset es el 4. Entonces lo que se hace es, se carga el contenido de R4 en R3. Y luego se le suma 4 bytes a la dirección apuntada por el registro 4. Osea que luego de que se ejecute en linea en R4 no esta mas la dirección anterior sino que va a estar dir+4.

ARM vector suma de pares

[2 ptos] Codificar un programa en assembler ARM de 32 bits que recorra un vector de enteros y genere un nuevo vector formado por elementos que resultan de sumar pares de elementos del vector original. Ej. vector original {1,2,5,6}, vector nuevo {3,11}

```
        .equ SWI_Print_Int, 0x6B
        .equ SWI_Exit, 0x11
        .equ SWI_Print_Str, 0x69
        .equ Stdout, 1
        .data
array_origen:
        .word 1,2,5,6
array_destino:
        .word 0, 0, 0, 0
array_length:
        .word 4
eol:
        .asciz "\n"
        .text
        .global _start
_start:
        ldr r0, =array_origen
        ldr r1, =array_destino
        ldr r2, =array_length
        ldr r2, [r2]
loop_suma:
        ldr r4, [r0]
        add r0, r0, #4
        sub r2, r2, #1
        ldr r5, [r0]
        add r6, r4, r5  @Si es resta se cambia la operacion por subs
        str r6, [r1]
        add r0, r0, #4
        add r1, r1, #4
        sub r2, r2, #1
        cmp r2, #0
        bne loop_suma
        ldr r2, =array_destino
        ldr r3, =array_length
        ldr r3, [r3]
loop_mostrar:
        cmp r3, #0
        beq exit
        ldr r0, =Stdout
```

```
        ldr r1, [r2]
        swi SWI_Print_Int
        ldr r1, =eol
        swi SWI_Print_Str
        add r2, r2, #4
        sub r3, r3, #1
        b loop_mostrar
exit:
        swi SWI_Exit
        .end
```

Clasi según número de direcciones

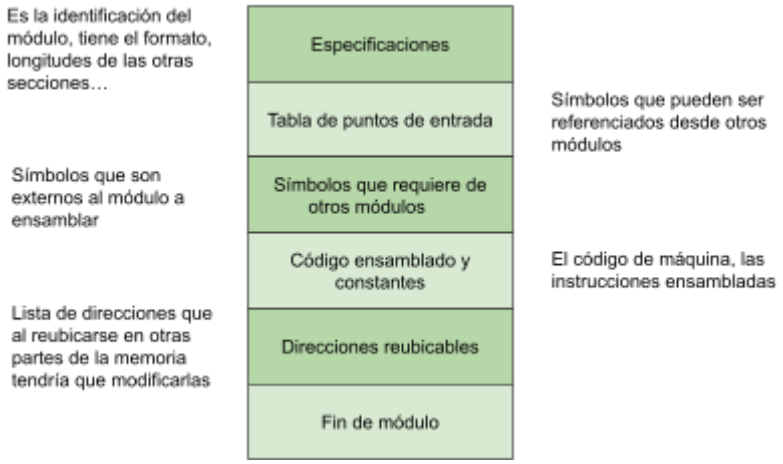
[1 pto] Indique como se puede clasificar el repertorio de instrucciones de una arquitectura de computadores de acuerdo al número de direcciones. Ejemplifique cada una.

Un repertorio de instrucciones de una arquitectura, puede clasificar según el numero de direcciones en:

- 0 direcciones: stack, por ejemplo: PUSH A, PUSH B, ADD
- 1 dirección: acumulador, por ejemplo: Abacus Load A, Add B, Store C
- 2 direcciones: registro/memoria (ADD R3,B), memoria/memoria (ADD A,B) o registro/registro (ADD R2,R1).
- 3 direcciones: registro/memoria (ADD R3,R1,R3) ej ARM

Código Objeto esquema

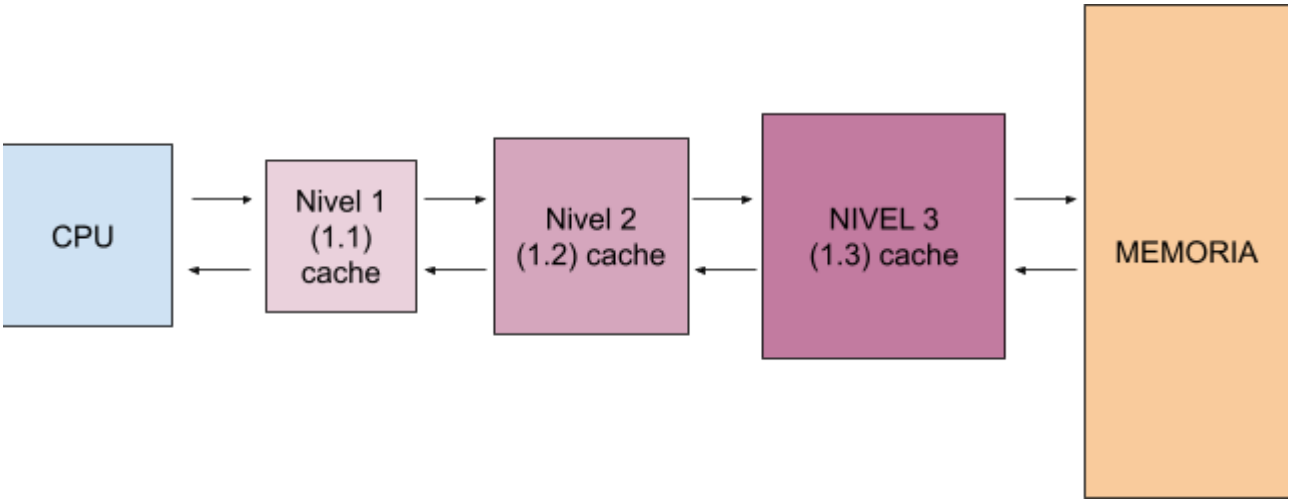
[1,5 ptos] Grafique el esquema general de un archivo de código objeto e identifique y explique cada una de sus secciones, indicando para que se usan.



Memoria caché

[1,5 ptos] En un sistema de memoria, ¿qué función cumple la memoria caché? ¿En qué principio se basa su efectividad? Grafique un ejemplo de arquitectura de cache de 3 niveles.

La memoria caché permite agilizar las búsquedas de datos en la memoria. Lo que hace es que cuando la CPU pide información a la memoria, en realidad primero le pide a la caché está si no tiene la palabra almacenada le pide a la memoria que se le da y también se lleva un bloque de palabras relaciones usando el principio de PRINCIPIO DE LOCALIDAD DE REFERENCIA, entonces la próxima vez que se le pida una palabra es probable que si la tenga guardada. La memoria caché es la más rápida y costosa de las memorias existentes. Con los años se fueron generando estructuras de niveles de cache, una de 3 niveles se veria asi:



Codificación 8-14(EFM)

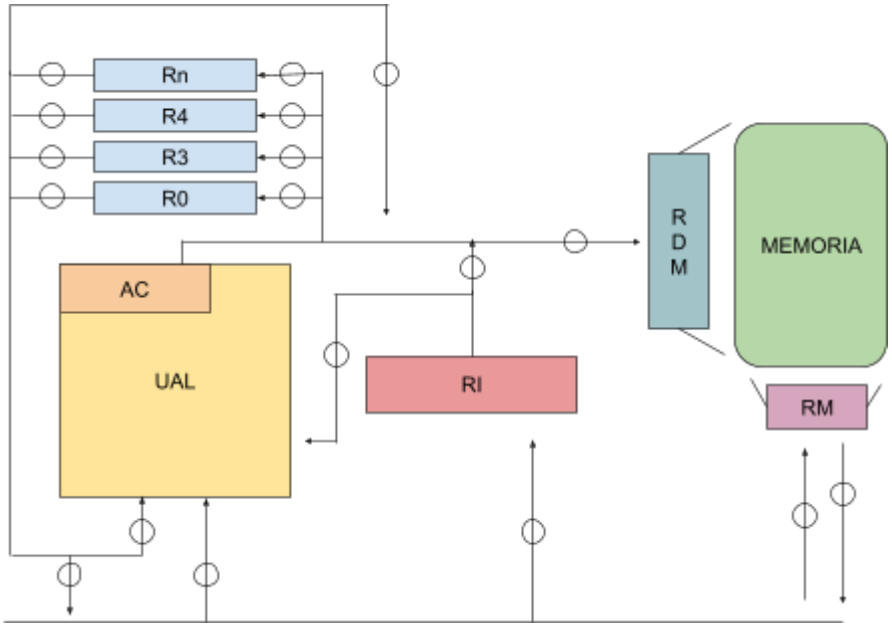
[1,5 ptos] ¿Qué es la codificación 8-14 (EFM) y para qué se usa? ¿Por qué es necesaria?

La codificación 8-14 (EFM) es un algoritmo que sirve como “traductor” para grabar datos en por ejemplo los discos opticos CD-ROM. Ya que los datos venían en 8 bits, y cada espacio para guardarlo en estos discos era de 14.

Final del 14/12/22

SuperAbacus operandos en memoria

[1,5 ptos] Explique claramente si es posible implementar en la maquina SuperAbacus la instruccion SUMAR DI(RI), DII(RII) donde DI, RI y DII, RII hacen referencia a dos operandos de memoria. Justifique su respuesta usando el gráfico de la máquina.



En la maquina SuperAbacus como la conocemos, no se podria llevar a cabo esa instruccion de forma directa, si se podria almacenar de forma momentanea en una registro extra la direccion DI(RI) final, para luego si operar de forma habitual y finalmente almacenar el resultado obtenido en DI(RI).

Modos de direc. en ARM

[1 pto] Explique claramente y ejemplifique al menos cuatro modos de direccionamiento presentes en la arquitectura ARM 32 bits. [Repetido]

ARM vector y and

[2 ptos] Codificar un programa en assembler ARM de 32 bits que recorra a un vector de enteros y genere un archivo de salida con el resultado de aplicar la funcion AND en cada uno de los elementos del vector original contra una constante.

Clasi según número de direcciones

[1 pto] Indique como se puede clasificar el repertorio de instrucciones de una arquitectura de computadores de acuerdo al número de direcciones. Ejemplifique cada una. [Repetido]

Linking dinámico en carga

[1,5 ptos] ¿Qué es el linking dinámico en tiempo de carga? ¿Qué ventajas tiene frente al estático? [Repetido]

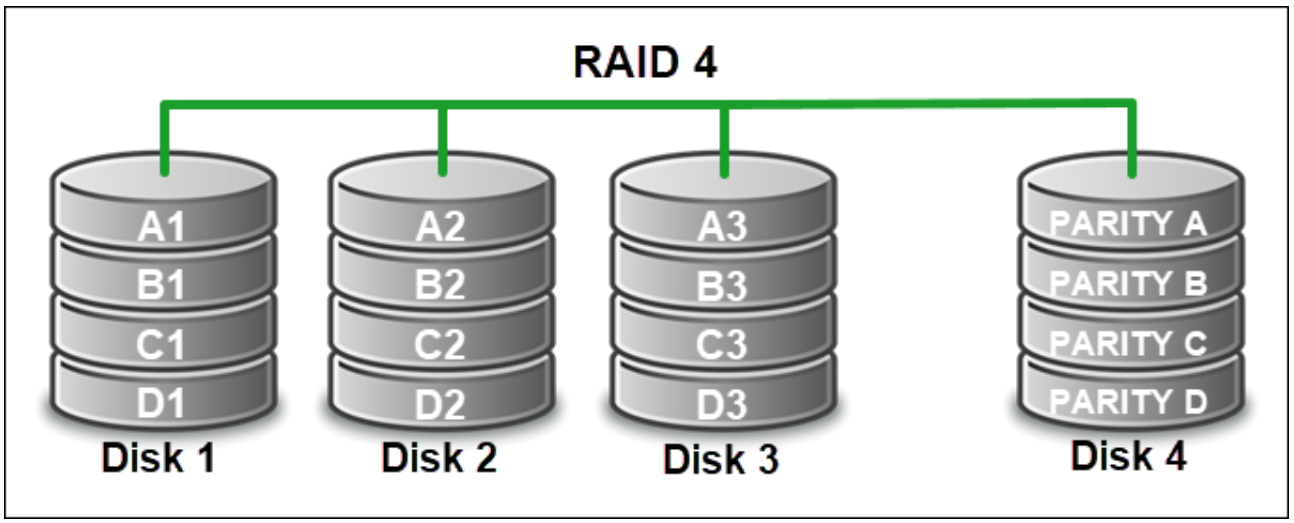
Memoria caché

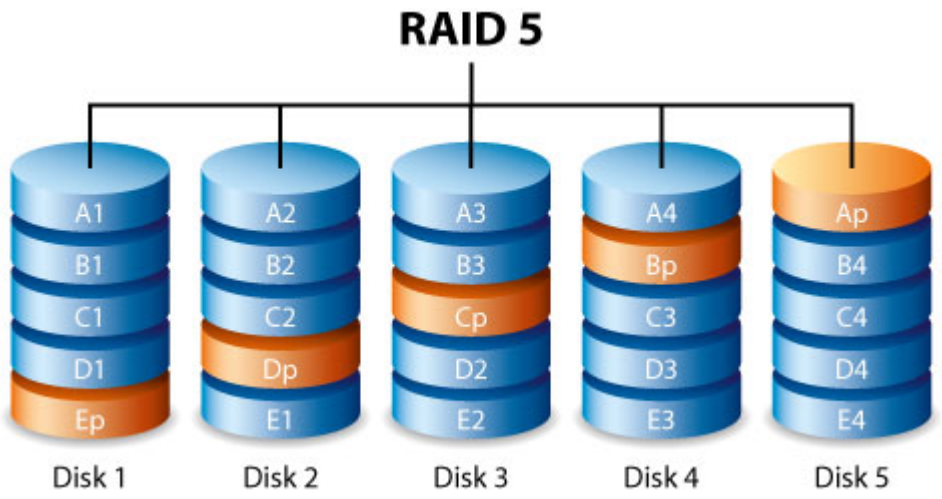
[1,5 ptos] En un sistema de memoria, ¿qué función cumple la memoria caché? ¿En qué principio se basa su efectividad? Grafique un ejemplo de arquitectura de cache de 3 niveles. [Repetido]

RAID 4 y 5

[1,5 ptos] ¿Cuáles son las ventajas del nivel 5 de la arquitectura de discos RAID con respecto al nivel 4? En qué caso lo usaría? Grafique la distribucion de la informacion en los discos en ambos niveles.

La arquitectura de discos RAID de nivel 5 requiere N+1 discos, y se accede de forma independiente a cada disco (hasta ahora lo mismo que el nivel 4), pero la novedad es que los strips de paridad se distribuyen en todos los discos resolviendo el cuello de botella que existia en el nivel 4, donde un disco tenia todos los strips de paridad. Esto igual tiene su desventaja ya que el controlador en el nivel superior es mas complejo.

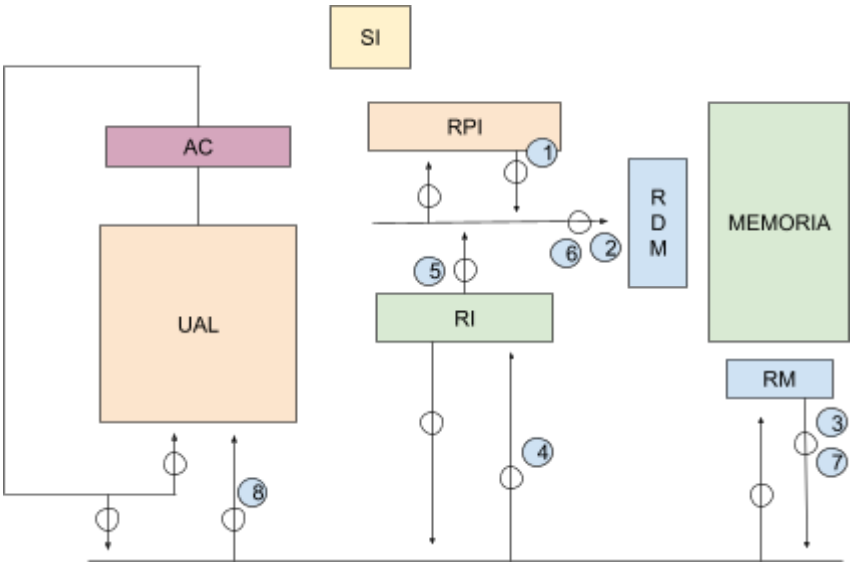



RAID 5
A1
B1
C1
D1
Ep
A2
B2
C2
Dp
E1
A3
B3
Cp
D2
E2
A4
Bp
C3
D3
E3
Ap
B4
C4
D4
E4
Disk 1
Disk 2
Disk 3
Disk 4
Disk 5

Final de 05/07/23

Abacus fases

[1,5 ptos] Indique cuales son las microinstrucciones necesarias para las fases de busqueda y ejecucion de la instruccion SUMAR 300 en la maquina Abacus, siendo 300 la direccion de una celda en base 16. Se pide ademas graficar en el esquema de la máquina el flujo de apertura de compuertas usadas en ambas fases de dicha instruccion.



Modos de direc. en ARM

[1 pto] Explique claramente y ejemplifique al menos cuatro modos de direccionamiento presentes en la arquitectura ARM 32 bits. [Repetido]

ARM vector y OR

[2 ptos] Codificar un programa en assembler ARM de 32 bits que recorra a un vector de enteros y genere un archivo de salida con el resultado de aplicar la funcion OR en cada uno de los elementos del vector original contra una constante. [Similar pero con AND]

Clasi según número de direcciones

[1 pto] Indique como se puede clasificar el repertorio de instrucciones de una arquitectura de computadores de acuerdo al número de direcciones. Ejemplifique cada una. [Repetido]

Linking dinámico en carga

[1,5 ptos] ¿Qué es el linking dinámico en tiempo de carga? ¿Que diferencias tiene frente al estático? [Repetido]

Jerarquia de memoria

[1,5 ptos] En un sistema de memoria, a que se denomina jerarquia de memoria? Indique claramente los elementos que la componen y las caracteristicas que identifican a cada uno. Grafiquela.

Se denomina jerarquia de memoria al orden que se le puede dar a las distintas formas de almacenamiento que hay. El orden va asi, arriba estan los mas rapidos, y a su vez los mas caros, tambien van de arriba hacia abajo los que menos a mas espacio tienen.

Registros	|	Estos tres son las unidades
Cache	|	de almacenamiento
Ram	|	interno
SSD	|	Estos ultimos son
Discos solidos	|	unidades de almacenamiento externos
CD/ DVD …	|	perifericos
Cintas magneticas	|	Tambien externas, se usan para back up

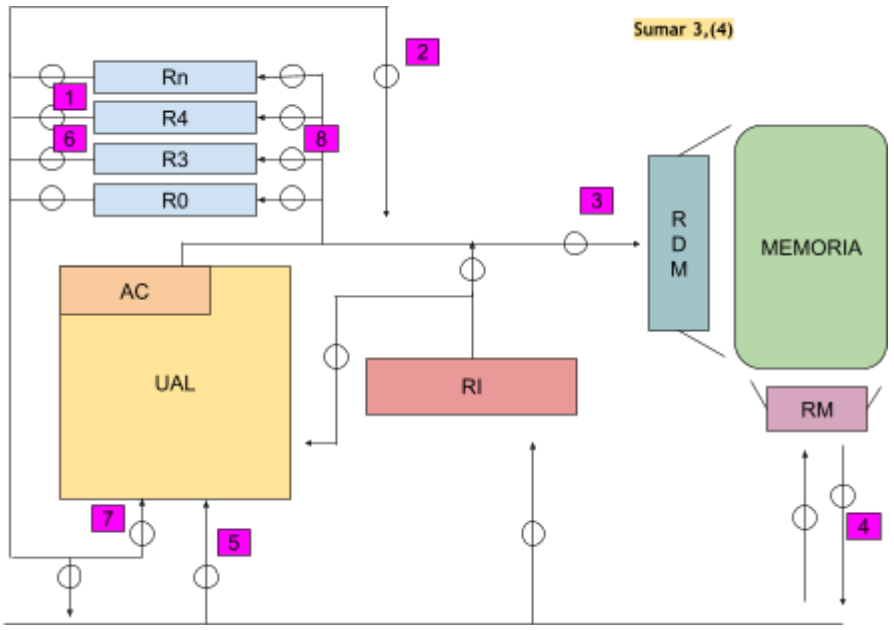
RAID 4 y 5

[1,5 ptos] ¿Cuáles son las ventajas del nivel 5 de la arquitectura de discos RAID con respecto al nivel 4? En qué caso lo usaría? Grafique la distribucion de la informacion en los discos en ambos niveles. [Repetido]

Final del 21/02/24

SuperAbacus sumar 3,(4)

[1,5 ptos] SuperAbacus de Sumar 3,(4)



(R3) -> AC
(R4) -> RDM
((RDM)) -> RM
(RM) -> AC
(R3) + ((R4)) -> AC
(AC) -> R3

Direccionamiento post-indexado

[1 ptos] Explique claramente que es y cómo funciona el modo de direccionamiento post-indexado autoindexado (registro indirecto con post-incremento) en la arquitectura ARM de 32 bits. De un ejemplo concreto con una instrucción. [Repetido]

ARM vector y and

[2 ptos] Codificar un programa en assembler ARM de 32 bits que recorra a un vector de enteros y genere un archivo de salida con el resultado de aplicar la funcion AND en cada uno de los elementos del vector original contra una constante [Repetido]

Clasificación del repertorio de instrucciones

[1 ptos] Indique como se puede clasificar el repertorio de instrucciones de una arquitectura de computadores de acuerdo con la ubicación de los operandos. Ejemplifique y/o grafique cada uno. [Repetido]

Código objeto esquema

[1,5 ptos] Grafique el esquema general de un archivo de código objeto e identifique y explique cada una de sus secciones, indicando para que se usan [Repetido]

Pipelining

[1,5 ptos] Definir el pipelinening, su desventaja y quién la debería resolver, y graficar un pipeline de 5 stages. [Repetido]

Codificación 8-14(EFM)

[1,5 ptos] ¿Qué es la codificación 8-14(EFM) y para que se usa?Por que es necesaria? [Repetido]

Final del 28/02/24

IEEE 754

[1,5 ptos] ieee 754 finalidad y gráfico de número desnormalizados

IEEE 754 nace como estándar de representación de números flotantes, ya que antes cada arquitectura tiene su forma de representarlo, con sus errores y también hacía que no sean compatibles entre sí. Por lo tanto a fines de los 70 aparece, buscando una generalización.

Uno de los problemas de esta representación es que tiene límites de mínimo y máximos representables. Generando así problemas de overflow o underflow. Para el primero no hay mucha solución más que dejarlo en el número máximo representable o pasar a un infinito positivo. Sin embargo, para los underflow hay una solución que es la representación de números desnormalizados. El formato es así:

1/0 | 00000000 | cualquier bit

esto agrega una rango de números que se pueden representar en este formato, sin embargo de excederse nuevamente,existe también el menos infinito.

Los infinitos operan como lo hacen en la matemática clásica. Por ejemplo algo más infinito es infinito.

Ejecución condicional ARM

[1 punto] En la arquitectura ARM de 32 bits, ¿a qué se denomina ejecución condicional de una instrucción? De un ejemplo de su uso en assembler [Repetido]

ARM lee enteros de archivo

[2 puntos] Codificar un programa en Assembler ARM de 32 bits que lea desde un archivo números enteros e imprima por la salida estándar la productoria de aquellos números que sean positivos.

Page fault

- [1,5 puntos] ¿Qué es un “page fault” y cuando ocurre? ¿Quién lo gestiona?

El “page fault” es un error esperado que arroja la memoria cuando se intenta acceder a una página (porción de proceso) que no está almacenada en la memoria. Este caso se da cuando la memoria es administrada de forma paginada por demanda, en este tipo de administración el proceso a almacenar en la RAM es dividido en páginas del mismo tamaño que los frames (nombre que recibe cada segmento de memoria), y solo algunos frames (los imprescindibles para que el proceso se ejecute) se guardan en la memoria, por ello es probable que durante la ejecución se requieran páginas no almacenadas.
Este error es controlado por el sistema operativo que genera una interrupción, y va a buscar a

la memoria virtual la página que falta, para luego almacenarla y que siga la ejecucion. Este proceso pasa muchas veces y puede provocar Trashing.

RAID 6 y 5

[1,5 ptos] ¿Cuáles son las ventajas del nivel 6 de la arquitectura de discos RAID con respecto al nivel 5? En qué caso lo usaría? Grafique la distribucion de la informacion en los discos en ambos niveles.

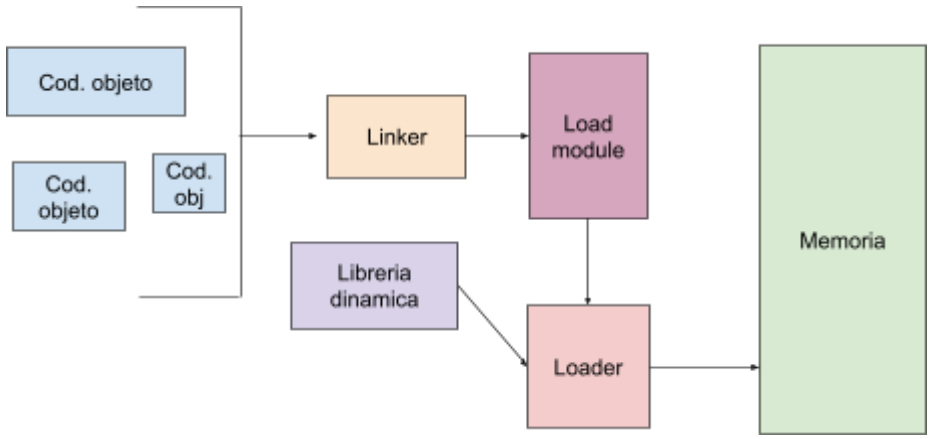
El raid de nivel 6 adiciona un disco de paridad más que en el nivel 5, lo que le da más velocidad en el procesamiento de los datos, permite que puedan romperse/dejar de funcionar hasta 2 discos de la arquitectura sin que se pierdan datos. La desventaja es que al tener un disco más es más costoso y al igual que en el 5 el cálculo de paridad es complejo.

block 0	block 1	block 2	block 3	P(0-3)	Q(0-3)
block 4	block 5	block 6	P(4-7)	Q(4-7)	block 7
block 8	block 9	P(8-11)	Q(8-11)	block 10	block 11
block 12	P(12-15)	Q(12-15)	block 13	block 14	block 15

Linking dinámico en carga

[1,5 ptos] ¿Qué es el linking dinámico en tiempo de carga? ¿Qué ventajas tiene frente al estático?

El linking dinámico en tiempo de carga es un tipo de linking que linkea con las referencias externas recién al momento en que se ejecuta el loader. Como ventaja frente al estatico, este pesa menos y facilita la actualizacion de versiones, ya que no es necesario volver a compilar.



Clasi según número de direcciones

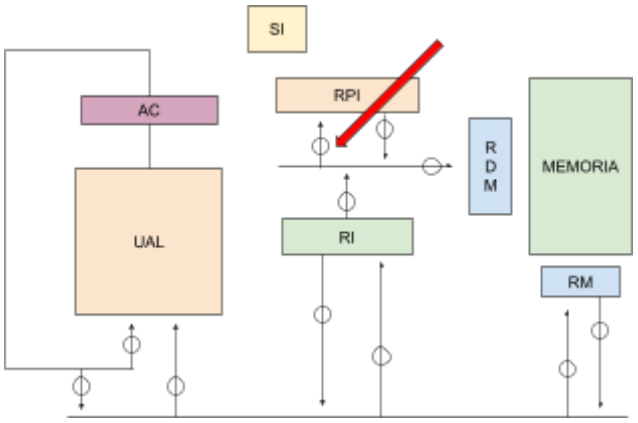
[1 pto] Indique como se puede clasificar el repertorio de instrucciones de una arquitectura de computadores de acuerdo al número de direcciones. Ejemplifique cada una. [Repetido]

Final del 06/03/24

Maquina Abacus

[1,5 ptos] Indique gráficamente en el esquema de la máquina Abacus cuál es la compuerta que permite que se cumpla el principio de ruptura de secuencia de Von Neumann. De un ejemplo de una instrucción en donde se aplique este principio.

El principio de ruptura de secuencias se cumple gracias a la compuerta que habilita que se guarde información el RPI, entonces por ejemplo en una bifurcación que se requiere desviar el orden de la secuencia, se puede cargar la siguiente instrucción directamente en el RPI provocando que sea la próxima instrucción a ejecutarse.



Direccionamiento post-indexado

[1 pto] Explique claramente qué es y cómo funciona el modo de direccionamiento post-indexado autoindexado (registro indirecto con post-incremento) en la arquitectura ARM de 32 bits. De un ejemplo concreto con una instrucción. [Repetido]

ARM vector suma de pares

[2 ptos] Codificar un programa en assembler ARM de 32 bits que recorra un vector de enteros y genere un nuevo vector formado por elementos que resultan de sumar pares de elementos del vector original. Ej. vector original {2,4,5,6}, vector nuevo {-1,1} [Repetido]

Big y Little Endian

[1 pto] Explique claramente qué significan los términos big y little endian, en qué contexto se aplican y qué los diferencia. De ejemplos de arquitecturas en donde se use cada uno.

Big vs Little Endian: se trata del orden de los bytes, Big Endian es el orden “natural” los bytes más significativos están en las direcciones más altas, y más abajo las menos signitifcativas. Ejemplos que usan este orden son IBM Mainframe. Por otro lado está Little Endian que es exactamente al revés, los bytes menos significativos están arriba en el orden de las

direcciones. Ejemplos que usan este orden son Intel x86. Y luego están quienes pueden usar ambos como es ARM, se debe configurar previamente.

Esto es relevante solo si debo manipular bytes de forma individual.

Segunda vuelta procesador

[1,5 ptos] Segunda vuelta del proceso de ensamblado

el proceso de ensamblado se puede llevara cabo en una o dos vueltas, primer se hace un pre-procesamiento, luego en una primera pasada se arma una tabla de simbolos y en la segunda pasada, se vuelve a recorrer todo el archivo de codigo ensamblador para, usando la tabla de simbolos hecha en la primer pasada, generar finalmente el codigo objeto.

Procesamiento en paralelo

[1,5 ptos] Las clasificaciones de procesamiento en paralelo de datos (son 2)

Son arquitecturas que implementan trabajar en una unica instrucción de máquina con multiples datos, existen dos formas de llevar a cabo esto:

* SIMD, se le aplica una unica instruccion a un conjunto grande de datos, es especialmente util para operaciones de grandes volumenes de datos como graficos, procesamiento de señales y aplicaciones cientificas.

* Vectorial, lo usan las intel core, similar a SIMD aplica una misma instruccion a un vector de datos, pero para eso los datos deben llegar en forma vectorial.

Almacenamiento en cinta

[1,5 ptos] Identifique y explique cuáles son las principales desventajas del medio de almacenamiento en cinta. ¿Cuáles son sus aplicaciones actuales? ¿Qué ventaja comparativa tiene con respecto al resto de los medios de almacenamiento secundario? [Repetido]

Examen Final 10/07/2024

Maquina SuperAbacus

[1,5 ptos] Explique cuáles son los modos de direccionamiento, formatos de instrucción y tipos de datos presentes en la máquina SuperAbacus. De ejemplos de cada uno de ellos.

Direccionamiento pre-indexado

[1 pto] Explique claramente qué es y cómo funciona el modo de direccionamiento pre-indexado autoindexado (registro indirecto con pre-incremento) en la arquitectura ARM de 32 bits. De un ejemplo concreto con una instrucción.

ARM vector “PAR”

[2 ptos] Codificar un programa en assembler ARM de 32 bits que recorra un vector de enteros y los imprima por la salida estándar agregando la leyenda "PAR" a continuación de todos aquellos números que así lo sean. [Repetido]

Big y Little Endian

[1 pto] Explique claramente qué significan los términos big y little endian, en qué contexto se aplican y qué los diferencia. De ejemplos de arquitecturas en donde se use cada uno. [Repetido]

Código objeto esquema

[1,5 ptos] Grafique el esquema general de un archivo de código objeto e identifique y explique cada una de sus secciones, indicando para que se usan. [Repetido]

Procesamiento en paralelo Superscalar

[1,5 ptos] Explique claramente de qué se trata la técnica de procesamiento en paralelo Superscalar en un procesador. Grafique en forma esquemática cómo funciona dicha técnica y dé un ejemplo de algún procesador comercial que la incluya.

La técnica de procesamiento en paralelo Superscalar en un procesador es una técnica que permite la ejecución de varias instrucciones al mismo tiempo. Para lograr esto, se utilizan varias técnicas avanzadas que permiten que las instrucciones se ejecuten en paralelo sin afectar la correctitud del resultado final.

La primera técnica utilizada en un procesador Superscalar es la emisión de múltiples instrucciones. En lugar de emitir una sola instrucción en cada ciclo de reloj, un procesador Superscalar es capaz de emitir varias instrucciones al mismo tiempo. Las instrucciones se emiten a un conjunto de unidades de ejecución que son capaces de procesar diferentes tipos de instrucciones al mismo tiempo.

La siguiente técnica utilizada es la renumeración de registros. En lugar de depender de un conjunto de

registros limitados, los procesadores Superscalar utilizan una técnica de renombrado de registros que les permite asignar temporalmente registros virtuales a las instrucciones. Esto permite que las instrucciones se ejecuten en paralelo sin afectar la correctitud del resultado final.

Hay varios procesadores comerciales que utilizan la técnica de procesamiento en paralelo Superscalar, incluyendo:

1. Intel Core i7 y i9: Estos procesadores de gama alta tienen múltiples núcleos y utilizan la técnica de procesamiento en paralelo Superscalar para mejorar el rendimiento de la CPU.
2. AMD Ryzen: Los procesadores Ryzen de AMD utilizan la técnica de procesamiento en paralelo Superscalar para mejorar el rendimiento de la CPU y ofrecer un mayor número de hilos.
3. IBM Power9: El procesador Power9 de IBM utiliza la técnica de procesamiento en paralelo Superscalar para mejorar el rendimiento y la capacidad de procesamiento en sistemas de alta gama, como servidores y supercomputadoras.

FALTA GRÁFICO

RAID nivel 3

[1,5 ptos] En la arquitectura de discos RAID de nivel 3: ¿qué ocurre con las peticiones de E/S mientras un disco queda inutilizable? [Repetido]

Examen Final 30/07/2024

Abacus y SuperAbacus PI

[1,5 ptos] Explique claramente cuáles son los pasos necesarios para actualizar el valor de la direccion de la proxima instruccion a ejecutarse en Abacus y SuperAbacus, en que se diferencian? especifique ademas las microinstrucciones necesarias en cada caso

Barrel Shifter

[1 pto]

ARM vector suma de pares

[2 ptos]

Intel como CISC

[1 pto]

Segunda pasada del proceso de ensamblado

[1,5 ptos]

Procesamiento en paralelo por multiprocesadores

[1,5 ptos]

RAID nivel 1 y 0

[1,5 ptos]

Extra de consignas que no se de que final son o me los invente

Unidad 1:

IEEE 754 manejo de infinitos

[1,5 ptos] ¿Qué mecanismos provee el estándar IEEE 754 para el manejo de números + - infinito? De ejemplos de dichas operaciones e indique cual sería la configuración en el formato para representar dichos resultados.

IEEE 754 para manejar los números +- infinitos propone una forma de representarlos:

1 | 11111111 | 00000000000000000000000 para el infinito negativo y

0 | 11111111 | 00000000000000000000000 para el infinito positivo.

Que los identifica de los normalizados porque en estos no está permitido que los 8 números del exponente sean 1s. Estos infinitos operan igual que lo hacen en las matemáticas por ejemplo algo + infinito daría infinito.

IEEE 754 manejo de indeterminados

[1,5 puntos] ¿Qué mecanismos provee el estándar IEEE 754 para el manejo de operaciones matemáticas con resultados indeterminados o indefinidos? De ejemplos de dichas operaciones e indique cual sería la configuración en el formato para representar dichos resultados.

En caso de resultados indeterminados o indefinidos IEEe 754 propone algunas soluciones.

Si es el caso que no se puede representar porque es menor al minimo representable de los numeros normalizados de la configuracion, se propone representarlos como numero desnormalizados, seria:

1 o 0 | 00000000 | cualquier patron de bits

este tipo de representacion soluciona algunos problemas de underflow que podian presentarse, mas si eso no alcanza, existen los infinitos, en este caso el negativo seria

1 | 11111111 | 00000000000000000000000

En caso de overflow, esta el infinito positivo. No existe una forma de representar mas numeros que los normalizados positivos.

0 | 11111111 | 00000000000000000000000

IEEE 754 extremos denormalizados

[1,5 puntos] Cuál es la finalidad de la existencia de los numeros desnormalizados en el formato IEEE 754? Grafique los valores extremos del rango de numeros desnormalizados y de sus configuraciones en hexadecimales en el formato.

La existencia de los numeros desnormalizados, es para poder representar numeros mucho mas pequeños que se escapaban del rango de los normalizados, y de otra manera quedarian en ceros. Esto logra poder expresar numeros que son utiles en el ambito cientifico y las ingenierias donde se requiere extrema precision.

Para representar numeros desnormalizados se sigue el siguiente formato:

+/- | 00000000 | cualquier serie de bits distinta de cero

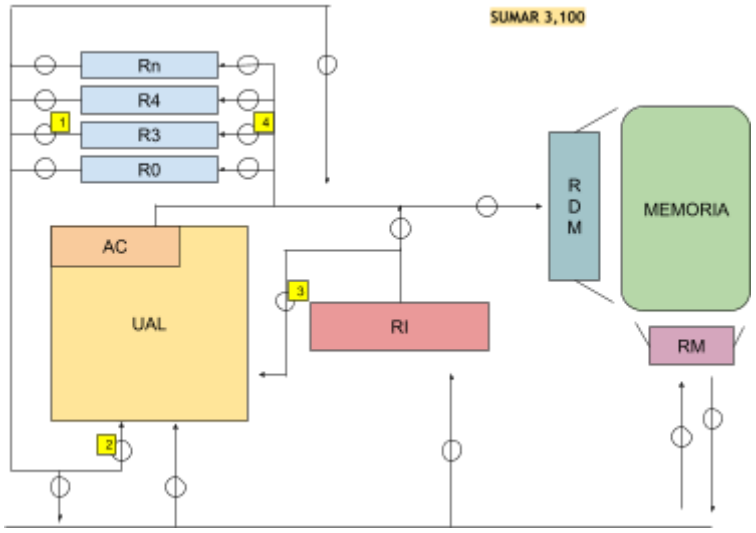
Entonces para representar los extremos tenemos

+ | 00000000 | 11111111111111111111

Unidad 2:

SuperAbacus de SUMAR 3,100

[1,5 puntos] Indique cuales son las microinstrucciones necesarias para ejecutar la instrucción SUMAR 3,100 en una maquina SuperAbacus, siendo 3 un registro de uso general y 100 un offset en base 10. Se pide además graficar en el esquema, el flujo de apertura de compuertas usadas en la fase de ejecución de dicha instrucción.



(R3) -> AC

(AC) + 100 -> AC

(AC) -> R3

Unidad 3:

SuperAbacus modos de dirrecionamiento

[1,5 puntos] Explique cuáles son los modos de direccionamiento presentes en la maquina Super Abacus. De ejemplos de cada uno de ellos.

Los modos de direccionamiento presentes en la máquina super abacus son:

- Inmediato: SUMAR 3,100
- Registro Directo: SUMAR 3,4
- Registro indirecto: SUMAR 3,(4)
- Base + desplazamiento: SUMAR 3,20(4)

4 elementos de ISA

[1 punto] Enumere por lo menos 4 elementos presentes en la arquitectura de programación (ISA) de un computador. De ejemplos de dichos elementos en alguna de las arquitecturas vistas en clase.

Algunos de los elementos presentes en la arquitectura de programación son:

- Repertorio de instrucciones (ej Intel: MOV, ADD)
- Tipo de datos (ejemplo Abacus: BPFlotante IEEe754)
- Memoria (ej Intel x86 word equivale a 32 bits)
- Modos de direccionamiento (por ejemplo en ABACUS es por STACK, en super abacus pueden ser 4 por registro directo, registro indirecto, inmediato o base + desplazamiento).
- Registros
- Formato de instrucciones (ej ABACUS es fijo 16 bits en total 4 para el cod. de operación y 12 para el operando)
- Especificación de su operación
- Control de flujo (Intel usa Condition Code)

ISA

¿Qué es la ISA? ¿Por qué los INTEL i3, i5, i9 son compatibles entre sí? y por que AMD es compatible con Intel?

La ISA, es la arquitectura del conjunto de instrucciones, es el conjunto de herramientas que se tiene a disposicion para dialogar con la maquina, esta compuesta por:

- Repertorio de instrucciones
- Especificacion de su operacion
- Registros
- Tipos de datos
- Modos de direccionamiento
- Formato de instrucciones
- Memoria

Los distintos Intel son compatibles entre sí porque todos comparten la misma arquitectura, siguen la arquitectura INTEL 64.

Por otro lado, Intel y AMD son compatibles porque se especifican de forma casi identica. Asi si yo quisiera correr un programa de INTEL, pudiera hacerlo en una AMD.

Organizacion de computadoras

¿A qué llamamos organizacion de computadoras?

Llamamos así a la implementación de la arquitectura, por ejemplo si hablamos de la instruccion de multiplicar, el nombre de la instruccion para llevarla a cabo está en la arquitectura de la compu, pero el cómo se lleva a cabo, eso está en la organizacion.
También se puede hablar de como por ejemplo la misma arquitectura intel se utiliza en diferentes organizaciones para poder armar distintos procesadores ya sea para abarcar mas mercado o para distintas funcionalidades, como puede ser uno para celulares, smathwatches o servidores.

Microarquitecturas

¿Cuál es la diferenciacion entre microarquitecturas cableadas y microprogramadas?

Se le llama microarquitectura, a los circuitos electronicos y puede ser cableada, que son todos componentes de hardware (por ejemplo los procesadores de tipo RISC, ARM), la otra forma es la microprogramada, donde hay componentes electronicos que tienen microprogramas (o software), tienen chips donde se almacenan micropasos (intel aun en ciertas arquitecturas complejas la utiliza)

Familia de computadoras

¿A qué llamamos familia de computadoras?

Son modelos con prestaciones y precios diferentes, que tienen la misma arquitectura base con distintas organizaciones (como lo es por ejemplo la familia INTEL 64, i3, i5, i9…)

Clasificacion de compus

¿Cómo se pueden clasificar las computadoras segun su poder de cálculo?

Las computadoras se pueden clasificar en:

- Supercomputadoras: son extremadamente rapidas, manejan altos volumnes de datos, poseen miles de CPU y están diseñadas para usos especificos.
- Macrocomputadoras o Mainframe: de uso comercial y cientificos, cientos de CPU, y manejan grandes volúmenes de datos, tienen muy alta disponibilidad (mainframe IBM es el mejor ejemplo)
- Microcomputadoras: las de uso doméstico en casa
- Computadoras portatiles
- Computadoras de mano: relojes inteligentes, celus

Categorías de intrucciones

¿En qué categorías podrian clasificarse las instrucciones de una arquitectura?

Las instrucciones se pueden clasificar en las siguientes categorías:

- Aritmeticas y logicas: add, divide, and, xor
- Movimiento de datos: move, load, store
- Entrada/Salida: start
- Control de flujo: jump, compare, branch

Tipos de operando

¿Qué tipos de operando conoce?

- Registro
- Memoria
- Inmediato

Control de flujo

¿A que se refiere con "control de flujo" y qué metodos conoce?

El control de flujo refiere a como en las distintas arquitecturas se puede manejar una condicion y una accion posterior a la condicion.

- Condition Code (CC): la usan Intel y ARM, basicamente en una instruccion de maquina (CMP) se evalúa y se deja info sobre que salió de la evalucion. Por ejemplo si era menor o mayor a cero. Y luego está la instrucción de bifurcacion que en base a los bits que dejó la de arriba, acciona o no.
- Condition Register: Esta es igual a la anterior, pero en lugar de que el resultado de la evalucion quede guardada en bits, queda guardado en un registro. Entonces la siguiente instruccion de bifurcacion en lugar de ver esos bits, va al registro.
- Compare and branch: Tanto la comparación como la bifurcacion estan en una unica instruccion.

Formato de instrucciones (Encoding)

¿Qué es el encoding y cuales son sus componentes y cómo pueden clasificarse?

El encoding define cómo se conforma cada uno de los bits de la instrucción de máquina. Cada arquitectura o familia de computadoras lo define a su manera. Generalmente está compuesto por el código de operación (siempre), operandos (pueden ser uno o más, cada uno con su tamaño), el modo de direccionamiento de cada operando (Intel), flags.

Se pueden clasificar en:

- Fijos: como es ARM, todas las instrucciones de máquina miden lo mismo.
- Variable: como es INTEL, tiene un unico formato pero no tiene un unico tamaño unico.

- Hibridos: como IBM Mainframe, hay muchos formatos, cada uno es fijo, pero hay distintas instrucciones con distintos formatos.

Modos de direccionamiento

¿Qué tipos de modo de direccionamiento hay?

- Inmediato: el operando está en la intrucción
- Memoria directa: en la instruccion tengo una direccion, y el operando está en la memoria (Abacus), desventaja que si el programa hay que moverlo, ya la instruccion seguro no es válida.
- Memoria indirecto: en la instrucción tengo una dirección que apunta a una dirección de memoria, ese lugar en la memoria, tiene otra dirección de memoria que apunta finalmente al dato. La ventaja que entrega por sobre la memoria directa es que la indirección nos permite movernos a cualquier parte de la memoria, ya que en la anterior por cuestiones logicas de lugar, no podría. La desventaja es que acceder a memoria es costoso, y en esta opcion se accede dos veces.
- Registro: En al instruccion se hace referencia a un registro, y ahi se encuentra el dato. (SuperAbacus)
- Registro Indirecto: en la instruccion se especifica un registro, pero dentro del registro hay una direccion de memoria, donde finalmente ahi está el dato. Es más lenta que la de solo registros, porque el acceso a memoria es costoso.
- Desplazamiento: Sirve para recorrer estructuras más complejas como una matriz. La direccion efectiva del operando surge de una suma,hay tres tipos distintos:
  Registro base: se suele especificar en la instruccion un registro y un offset. Entonces se suma la direccion contenida en el registro más el offset para llegar a la direccion que voy a buscar en memoria.
  Indexado: que es igual pero al revés, en la instrucción está la direccion y en el registro está el offset.
  Relativo (program counter): sirve para desplazarse pero desde la ubicacion donde esta la instruccion que se esta leyendo al momento. Es jugado porque depende de que no se agreguen o quiten instrucciones en el medio de la que estoy y la que quiero llegar para que el offset no quede mal.
- Stack: No se especifica ningun operando, por ej: ADD, se entiende que está implicito.

Memoria de una máquina

¿Cuáles son las caracteristicas que describen a una memoria?

Se puede caracterizar segun su:

- Direccionamiento

- Tamaño de palabra (word size), es igual a los bits de la arquitectura. Sirve para la reserva de espacio. En Intel el concepto de word está asociado a 16 bits, por la primer máquina.
- Big vs Little Endian: se trata del orden de los bytes, Big Endian es el orden “natural” los bytes más significativos están en las direcciones más altas, y más abajo las menos signitifcativas. Ejemplos que usan este orden son IBM Mainframe. Por otro lado está Little Endian que es exactamente al revés, los bytes menos significativos están arriba en el orden de las direcciones. Ejemplos que usan este orden son Intel x86. Y luego están quienes pueden usar ambos como es ARM, se debe configurar previamente.
  Esto es relevante solo si debo manipular bytes de forma individual.
- Espacio de direcciones (adress space): Es para entender el tamaño que va a ocupar el programa.

Unidad 4:

3 modos de direccionamientos de Intel

[1,5 puntos] Mencione al menos 3 modos de direccionamiento presentes en la arquitectura Intel x86 dando un ejemplo de uso de cada uno en una instrucción.

Tres ejemplos de modos de direccionamiento en la arquitectura Intel x86:

- Implícito: el dato está implicito en el código de operación
  ej: CBW
- Registro: el dato está en un registro
  ej: mov RAX, 5
- Inmediato: el dato está dentro de la instrucción
  ej: mov RAX, 5
- Directo: el dato está en memoria referenciado por el nombre de un campo
  ej: mov RAX, [variable]
- Registro Indirecto: el dato está en memoria apuntado por un registro base o índice
  ej: mov EAX, [EBX]
- Registro relativo: El dato está en la memoria apuntado por un registro base más un desplazamiento
  ej: mov RAX, [RBX+4
- hay 3 más pero son medios raros..

Lenguaje Ensamblador

¿Por qué usar o aprender hoy en día lenguaje ensamblador? ¿De que está compuesto?

El lenguaje ensamblador se utiliza para el desarrollo de compiladores y de sistemas embebidos, para optimizar código en tamaño o velocidad, para trabajar con drivers de hardware, para el acceso a instrucciones que no están disponibles en alto nivel.
Está compuesto por etiquetas, mnemonicos, operandos y comentarios.

Ensambladores

¿Cómo sucede el proceso de traduccion en un ensamblador?

Los ensambladores son programas que traducen los programas escritos en lenguaje ensamblador y generan lenguaje de máquina, específicamente, saca un archivo que se denomina código objeto.
El funcionamiento del ensamblado implica “dos pasadas”. Consiste en que precisa leer entero dos veces el archivo de texto en código Assembler.
Lo primero que se hace es un pre-procesamiento, y luego se pasa a la primera pasada que genera una tabla de simbolos con todas las etiquetas y la ubicación donde se definió, y esa ubicacion esta definida segun el location counter que arranca en 0, lo que busca es calcular la longitud de la instrucción de máquina. (en ARM esto no es necesario porque ya está definido el tamaño de instrucción).
En la segunda pasada se hace la traduccion, se pasa de nuevo por todas las sentencias y se traduce aquellas que fuera necesario, como lo son los mnemonicos, cada nombre de operando en el registro. Cuando encuentra en una instruccion una etiqueta que está siendo usada, la traduce a un valor apropiado de LC usando la tabla de simbolos que hizo en la primera pasada.

INTEL (tipos de datos y memoria)

¿Cuáles son los tipos de datos que admite la arquitectura INTEL? ¿Cuáles son las caracteristicas de su memoria?

La arquitectura acepta tipos de datos numericos enteros (BPF c/s), numericos decimales (IEEE 754) y caracteres (ASCII).
En cuanto a su memoria, tiene celdas de 1 byte, word size de 2 bytes.
El metodo que utiliza para almacenar datos mayores a un byte es Little-Endian, es decir, el dato final está en la direccion de menor valor.

Primer pasada de ensamblador

Indique claramente cuales son las acciones que realiza un ensamblador específicamente en la primera pasada de proceso de ensamblado. ¿Qué informacion genera como salida dicho proceso y para que se usará?

Unidad 5:

Acceso secuencial de memoria

[1,5 puntos] Explique claramente cuáles son las características del modo de acceso secuencial y en qué tipo de memoria está presente.

El modo de acceso secuencial está presente en memorias del tipo cintas magnéticas por ejemplo, se caracteriza por almacenar datos uno detrás del otro sin dejar espacios, de forma lineal.

Este tipo de almacenamiento es DE TIEMPO VARIABLE a la hora de buscar información porque para encontrar un dato hay que pasar por todos los anteriores. La unidad de datos son los registros (records).

Acceso asociativo de memoria

[1,5 puntos] Explique claramente cuáles son las características del modo de acceso asociativo y en qué tipo de memoria está presente.

Este modo de acceso es el que usan las memorias cache. Se caracteriza por:

- Acceden de forma aleatoria por comparación de patrón de bits.
- Cada posición de memoria tiene un mecanismo de direccionamiento propio.
- La palabra se busca por una porción de su contenido en vez de por su dirección.
- Tiempo de acceso constante.

Acceso directo de memoria

[1,5 puntos] Explique claramente cuáles son las características del modo de acceso directo y en qué tipo de memoria está presente.

El modo de acceso directo es el que utilizan los discos magnéticos.

Se caracterizan por tener un tiempo de acceso variable, y porque cada bloque/registro tiene una dirección física única.

Acceso Aleatorio de memoria

[1,5 puntos] Explique claramente cuáles son las características del modo de acceso aleatorio y en qué tipo de memoria está presente.

El modo de acceso aleatorio es el que utiliza la memoria principal y algunas memorias caché.

Se caracterizan por tener un tiempo de acceso constante y porque cada posición de almacenamiento en la memoria se puede acceder mediante un mecanismo de cableado.

Almacenamiento en cinta

[1,5 puntos] Identifique y explique cuáles son las principales desventajas del medio de almacenamiento en cinta. ¿Cuáles son sus aplicaciones actuales? ¿Qué ventaja comparativa tiene con respecto al resto de los medios de almacenamiento secundario?

Las cintas magnéticas son usadas actualmente como back up de información. Sus principales desventajas son su almacenamiento secuencial que hace que cada vez que se quiera acceder a un punto en específico haya que recorrer para encontrarlo, por lo tanto es muy lento. Su ventaja es la gran capacidad, el bajo costo y la gran vida útil que tiene.

Parámetros de performance en memorias

[1,5 puntos] ¿Cuáles son los parametros de performance en un sistema de memoria?

Hay dos parámetros para medir la performance de un tipo de memoria:
* Tiempo de acceso (latencia): en las memorias con acceso aleatorio es el tiempo que se demoria en leer y escribir, en cambio en las que no tienen acceso aleatorio, es el tiempo que demora en llegar a la posicion, es decir estar listo, para escribir o leer.
* Tiempo de ciclo de memoria: es el tiempo de acceso más el tiempo que se toma la unidad de memoria para estar lista para comenzar otra operacion.
* Tasa de transferencia: tasa con la cual los datos son transferidos dentro o fuera de la unidad de memoria.

Principio de localidad de referencia

[1,5 puntos] ¿Qué es el principio de localidad de referencia?

Es un proceso que ocurre al ejecutar cualquier programa, que permite que las memorias caché sean efectivas, sin este no servirían. El principio anuncia que cada vez que se ejecuta un programa, las referencias a memoria que hace el procesador, es decir, cada vez que una instruccion quiere acceder a la memoria o para datos, tienden a estar agrupadas.

Interrupciones

[1,5 puntos] ¿Para qué existen las interrupciones? ¿Qué es lo que tratan de mejorar?

Las interrupciones llegan para intentar mejorar el rendimiento de los procesadores. Estas lo que hacen es interrumpir el proceso en el que está el procesador en ese momento para “avisarle” que otra cosa que necesitaba antes ya está lista o requiere ser atendida. Gracias a esto, el procesador puede por ejemplo hacer un pedido a un periférico como una impresora y no perder tiempo esperando la respuesta, sino que sigue con sus tareas y espera la interrupción que genera el módulo de I/O para continuar.

Canales E/S funcion

[1,5 puntos] ¿Qué ventajas otorgan los canales de E/S? ¿Qué funciones cumple el canal y que la CPU?

Los canales de entrada y salida le facilitan la comunicación con los periféricos al CPU. Es una forma de esconder las diferencias que tienen por ejemplo en los tiempos, la CPU es mucho mas rapida que el tiempo de respuesta que tienen los periféricos, entonces los canales de E/S actúan de intermediario para llevar los tiempos y por medio de interrupciones comunicar al CPU cuando sus tareas están listas.

Tecnica DMA de E/S

[1,5 puntos] Explique claramente qué ventajas aporta la tecnica de E/S por DMA frente a otras más limitadas. Grafique algunas posibles topologias de configuracion.

Funciones de módulo E/S

[1,5 puntos] Explique claramente al menos 3 funciones de los módulos de E/S.

Tres funciones de los módulos de E/S:

- Comunicación con el CPU
- Comunicación con los periféricos
- Control de los tiempos
- Detección de errores

Módulo de E/S

[1,5 puntos] Nombre al menos tres causas por las cuales es necesaria la existencia de los módulos de E/S para la interconexión de periféricos con el resto del sistema.

La existencia de módulos E/S es necesaria para poder manejar los tiempos y tipos de datos diferentes que maneja el procesador contra los que manejan los periféricos. Estos módulos se encargan de:

- Comunicarse con el CPU
- Comunicarse con los periféricos
- Recibir errores
- Organizar los tiempos distintos que manejan entre sus comunicadores.

Administracion de memoria paginada

[1,5 puntos] ¿Qué ventajas provee la administración de memoria paginada frente a otros mecanismos más sencillos? ¿Qué desventaja presenta frente a la administración paginada por demanda?

La administracion de memoria paginada aparece para dar una ventaja sobre la cantidad de memoria que se perdia en las administraciones antecesoras. Antes los procesos se insertan en memoria uno atras del otro, cada uno con sus tamanios pero esto solo era optimo la primera vez que se ordenaban, luego cuando uno terminaba, dejaba un hueco entre proceso y proceso de un tamanio X y si habia uno nuevo que igresar de tamanio menor iba para ahi dejando huecos de memoria libre sin aprovechar. La administracion de memoria página propone dividir cada proceso en páginas, y los espacios de memoria en frames, tanto página como frame median los mismo entonces las páginas se ubicaban en los frames disponibles sin necesidad de que uno esté correlativo al otro del mismo proceso. Esto lograba que los procesos sean rapidos ya que siempre estaba todo disponible en memoria pero también habia aun memoria usando sin necesidad porque no se usaban todas las páginas que se cargan en memoria. Es por esto que nace la administracion paginada por demanda, donde solo se cargan en memoria las paginas necesarias para que se ejectute el proceso, no todo, esto soluciono el no tener memoria ocupada que no se usa.

Trashing

[1 punto] Explique claramente qué es el fenómeno de “thrashing” y qué lo puede originar.

El fenomeno de trashing es cuando la computadora se pone lenta, y se origina cuando la computadora usa como sistema de administracion de memoria la paginacion por demanda, entonces lo que pasa es que los procesos se divididen en páginas, y la memoria en frames. Cada frame mide lo mismo que una página, entonces un proceso almacena en la memoria las páginas necesarias para que se ejecute el proceso, no todas, la que no las deja en memoria virtual, por lo tanto cuando alguna de esas paginas que si quedaron en la RAM desea acceder a algo que no lo está, el sistema operativo tiene que iora buscar la página a la memoria virtual y cargarla en RAM, ese proceso se hace muchas veces y para cada ejecucion que está realizando la CPU a la vez.

Discos SSD vs Duros Mecanicos

[1,5 puntos] Mencione al menos 4 ventajas de los discos SSD frente a los discos duros mecánicos.

Ventajas de los discos SSD sobre los discos duros mecánicos:

- Son más rápidos en el acceso a datos.
- No hacen ruido.
- No levantan temperatura.
- Consumen menos energía.
- Resisten golpes.
- Mayor seguridad al borrar datos.

Metodos de acceso a memoria

[1,5 puntos] Describa las características de los Métodos de acceso de unidades de datos directo y aleatorio.

El método de acceso directo:

- Tiene un tiempo de acceso variable.
- Cada registro/bloque está identificado por una dirección unica.
- Es el método que usan los discos magnéticos.

El método de acceso aleatorio:

- Tiene un tiempo de acceso fijo.
- Cada posición de memoria es accedida mediante mecanismos de cableado físico.
- Es el método que usan las memorias principales y algunas cache.

Almacenamientos externos

[1,5 puntos] Hacer un cuadro comparativo de los componentes de almacenamiento externos, dar características, ventajas y desventajas de c/u

Almacenamiento	Tamaño de almacenamiento	Velocidad	Costo	Otras características
Memoria SSD	D: Menor que el	V: La mas	D: El más alto	V: No levanta

	de un disco magnético.	rapida de las externas	de las externas	temperatura, ni hace ruido. V: Consume poca energía. D: Menor vida útil que los otros.
Disco Magnéticos	Mayor capacidad que SSD pero menos que cintas.	Es menor a la velocidad de la SSD pero mejor que los otros.	Menor que el de la SSD pero mayor que los inferiores.	D: Hacen ruido. D: Genera mucho calor.
Dispositivos ópticos	D: La capacidad de almacenamiento es poca.	V: Es bueno para la transferencia de datos.	V: Es barato a comparación de la SSD y el disco magnético.	Se usan para backup. V: Son portables.
Cintas magnéticas	V: Alta capacidad de almacenamiento	D: Son las más lentas	V: Son las más baratas	V: Duran mucho tiempo.

Procesadores CISC y RISC

[1 punto] Enuncie al menos 4 características de la arquitectura de procesadores CISC.

Los procesadores CISC como lo es Intel x86 son:

- Orientados al hardware
- Tienen un repertorio de instruccion muy amplio
- Las instrucciones son complejas
- Microarquitectura programada
- Muchos tipos de datos
- Pocos registros de uso especifico
- Compiladores simples
- Muchos mecanismos de direccionamiento

Los procesadores RISC como lo es ARM son:

- Orientados al software
- El repertorio de instrucciones es chico
- Las instruccion son de tamanio fijo (ARM 32 bits)
- Muchos registros de uso general
- Pocos tipos de datos
- Microarquitectura por cable
- Compiladores complejos

Interrupciones multiples

[1,5 puntos] Explique claramente los mecanismos para atender múltiples interrupciones. Explique las ventajas y desventajas de cada método.

Existen dos mecanismo para atender multiples interrupciones.

- Inhibicion de interrupciones. Aca si está lidiando con una interrupción, deja la nueva en stand by hasta que resuelva en lo que estaba, este manejo es simple pero su desventaja es que no tiene en cuenta que hay interrumpciones que son más urgentes que otras.
- Por prioridad. Aca atiende las interrupciones segun el orden de prioridad de una tablita.

Procesamiento Multithreading

Explique claramente de que se trata la tecnica de procesamiento en paralelo multithreading en un procesador. Explique sinteticamente la diferencia entre un thread y un proceso, y como es el cambio de contexto en un caso y otro.

Unidad 6:

Organizacion de discos magneticos

[1,5 puntos] Explique claramente cuáles son las ventajas y desventajas de la organización tradicional de discos magnéticos versus la organización multi zona. Grafique ambas organizaciones.

La organizacion tradicional de discos magneticos tiene como ventaja que se puede referencia a cualquier sector del disco, cosa que en el multizona es más complicado. Su desventaja es que tiene menos almacenamiento que un multizona, porque en los tradicional entra la misma cantidad de datos en la zona interior y en la zona exterior del disco.

[1,5 puntos] ¿Cuáles son las ventajas del nivel 1 de la arquitectura de discos RAID respecto al nivel 0? Grafique la distribución de la información en los discos en ambos niveles.

En el nivel 0 no hay forma de rescatar datos si se rompen los discos, en el nivel 1 hay un duplicado por cada disco entonces como ventaja puede dejar de funcionar alguno y que siga siendo util la arquitectura, tambien es mas rapido porque hay doble acceso al mismo dato si no hay ninguno disfuncional. La desventaja del de nivel 1 es que es el doble de caro.

[1,5 puntos] ¿Cuáles son las ventajas del nivel 5 de la arquitectura de discos RAID con respecto al nivel 4? Grafique la distribución de la información en los discos en ambos niveles.

Ventajas - Distribuye las tiras de paridad a lo largo de todos los discos. Lo cual evite un potencial cuello de botella de E/S encontrado en el RAID 4.

Otros:

[1,5 puntos] Explique claramente cuáles son los eventos temporales presentes a la hora de almacenar o recuperar información en un disco magnético sectorizado. Especifique como haría el cálculo de lectura de un archivo con una distribución aleatoria de la información en el disco. Ejemplifique de ser necesario.

- Tiempo de seek: tiempo que tarda la cabeza movil en posicionarse en la pista
- Latencia rotacional: una vez en la pista es el tiempo que toma en rotar hasta llegar a la posicion.
- Tiempo de acceso: la suma de los tiempos anteriores.
- Tiempo de transferencia de datos: Lo que se tarda en hacer la operacion de lectura/escritura.

Inventados:

En superAbacus ¿se pueden hacer operaciones entre dos operandos que estén en memoria?

No tiene respuesta unica, si está bien justificado pueden ser ambas, la más logica desde la coherencia del diseño electronico que podria funcionar pero forzado. Como la dan en el curso siempre la máquina tiene como primer operando un registro y del otro lado puede ir un area de memoria. El caso que se plantea es modificando el RI y habilitando que tenga los componentes necesarios para que se banque tener dos operandos de memoria. Entonces asi como esta, la maquina no lo bancaria. Peeero lo que si podria hacer es meter el dato de memoria momentaneamente en un registro cualquiera como auxiliar.