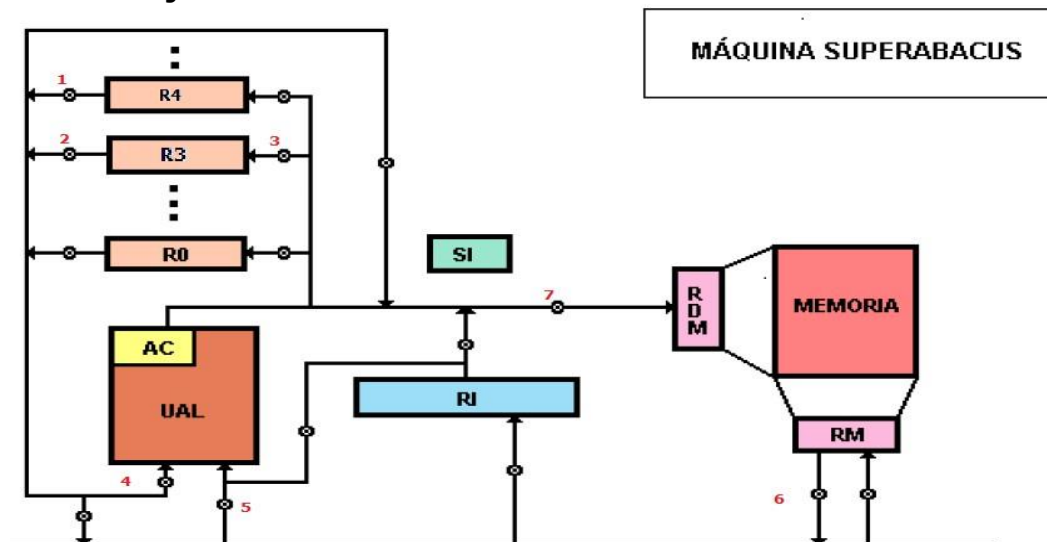


75.03 organización del computador

- [1,5 puntos] Indique cuales son las microinstrucciones necesarias para ejecutar la instrucción SUMAR 3,20(4) en una maquina Superabacus, siendo 3 y 4 registros de uso general y 20 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.



AC	←	(R4)	1,4
AC	←	(AC)+20	
RDM	←	(AC)	7
RM	←	((RDM))	
AC	←	(RM)	6,5
AC	←	(AC)+(R3)	2,4
R3	←	(AC)	3

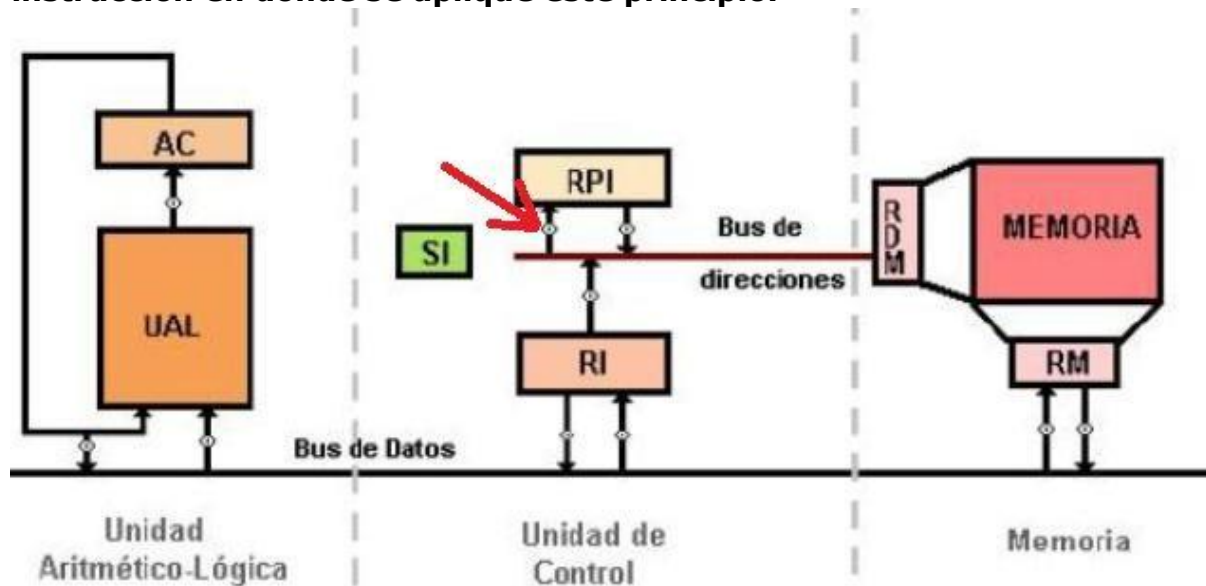
- [1,5 puntos] Indique cuales son las microinstrucciones necesarias para ejecutar la instrucción SUMAR 3,100 en una maquina SuperAbacus, siendo 3 un registro de uso general y 100 un offset en base 10. Se pide además graficar en el esquema, el flujo de apertura de compuertas usadas en la fase de ejecución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

Figura de Abacus anterior

AC	←	(R3)	1,4
AC	←	(AC)+100	
R3	←	(AC)	3

- [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.



La instrucción es la de salto:

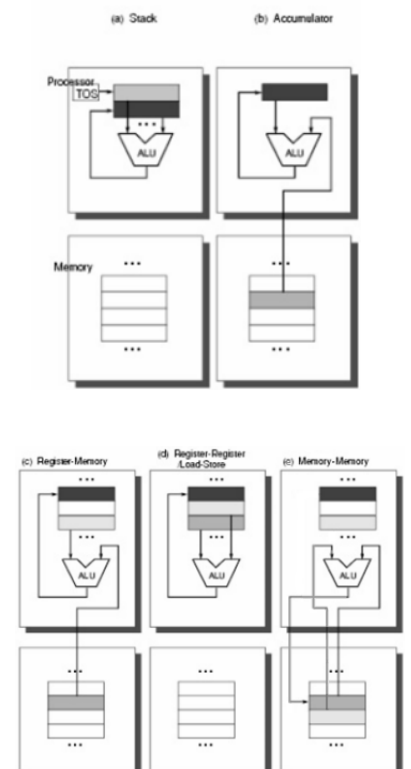
$RPI \leftarrow (OP)$

Se debe "saltar" a la dirección indicada en la instrucción. La dirección de bifurcación debe ser transferida al RPI (Para buscar la próxima instrucción). Esto rompería con la secuencialidad del programa y saltaría a otro sector de código. Esto cumpliría con el principio de Von Neumann de ruptura de secuencia, la cual es la clave del poder decisorio de los programas.

- **[1 punto] 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.**

$C = A + B$

- Stack (Pila)
 - PUSH A
 - PUSH B
 - ADD (Toma operandos de la pila)
 - POP C
- Acumulador
 - Load A (en el acumulador)
 - ADD B (Acumulador implícito)
 - STORE C (Guarda el contenido del acumulador en memoria)
- Registro-Memoria
 - Load R1, A (en registro R1)
 - ADD R3, R1, B (Suma B con R1 y guarda en R3)
 - STORE R3, C (en memoria)
- Registro-Registro
 - Load R1, A
 - ADD R2, B
 - STORE R3, R1, R2
 - Store R3, C
- Memoria
 - MOVE C, A
 - ADD C, B



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

- Tipo de acceso aleatorio por comparación de patrón de bits.
- La palabra se busca por una porción de su contenido en vez de por su dirección.
- Cada posición de memoria tiene un mecanismo de direccionamiento propio.
- Tiempo de acceso constante, independiente de las secuencias de accesos anteriores o su ubicación.

Este modo de acceso está presente en las memorias caché. Estas memorias guardan los últimos bloques usados por el CPU. Este le pide una palabra y esta es buscada en la caché, si llegara a estar devuelve la palabra indicada. Si no se busca el bloque, en el que esta palabra se encuentra, en memoria principal. Luego se guarda el bloque en la caché y se entrega la palabra deseada por el procesador.

- **[1,5 puntos] En la arquitectura de discos RAID de nivel 3: ¿Qué ocurre si un disco queda inhabilitado? ¿Cómo se puede recuperar la información perdida?**

Este nivel de la arquitectura de discos RAID consiste en N discos de datos más uno de paridad.

La paridad se calcula con un sencillo bit de paridad para conjuntos de bits individuales en la misma posición en todos los discos de datos.

La reconstrucción de datos es bastante sencilla considerando 5 discos, de los que de X_1 a X_3 contienen datos y el X_4 es el de paridad. La paridad del *iésimo* bit se calcula de la siguiente forma:

$$X_4(i) = X_3(i) + X_2(i) + X_1(i) + X_0(i)$$

Siendo + la función or exclusivo

Su poniendo que fallo la unidad $X_1(i)$. Si sumamos $X_1(i) + X_1(i)$ a los dos miembros de la ecuación, nos quedaría:

$$X_1(i) = X_4(i) + X_3(i) + X_2(i) + X_0(i)$$

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

Las interrupciones son un mecanismo por el cual un módulo puede interrumpir al procesador en lo que estaba haciendo para que pase a hacer otra cosa. Existen para mejorar la performance global del procesador y la eficiencia en el procesamiento. Si no existieran, el CPU se quedaría esperando y recién ahí seguiría ejecutando. También sirven para avisarle que puede retornar un proceso al CPU. Como ejemplo concreto, las interrupciones de E/S logran maximizar el uso del procesador

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

Los modos de direccionamiento presentes en una máquina SuperAbacus son:

- Inmediato

SUMAR	R4		100
-------	----	--	-----

- Registro Directo

SUMAR	R4	R5	
-------	----	----	--

- Registro indirecto

SUMAR	R4	R5	
-------	----	----	--

- Base + Desplazamiento

SUMAR	R4	R5	100
-------	----	----	-----

- **[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.**

- Repertorio de instrucciones (instrucciones de maquina).
- Registros (Cantidad, tipo, tamaño).
- Formatos de instrucción.
- Modos de direccionamiento.
- Memoria
 - o Espacio de direcciones.
 - o Tamaño de la celda.
 - o Formato de palabra.
- Tipos de datos.
- Tipos de operandos.
- Especificaciones de la operación de las instrucciones.
- Interrupciones.

- **[1,5 puntos] ¿Qué es un "page fault" y cuando ocurre?**

En el sistema de administración de memoria paginada por demanda, se van subiendo las páginas que se necesitan. Cuando el CPU pide una página que no está cargada en memoria principal se produce una interrupción llamada "Page Fault". Esta interrumpe el proceso y provoca que el sistema vaya a buscar la hoja deseada a memoria secundaria.

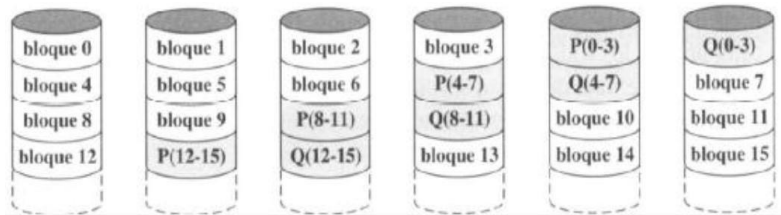
Un algoritmo de asignación ubica inicialmente algunas páginas del programa en la memoria. Por cada requerimiento de acceso a una página se consulta la tabla para saber en qué bloque esta. Si la página no está en memoria, se demanda la carga de esa página en memoria. Se detiene el procesamiento y genera automáticamente una interrupción, esta interrupción se conoce como interrupción de página (page fault). Es entonces el sistema operativo el que ejecuta la función de acceder al almacenamiento secundario para adquirir la página pedida.

- **[1,5 puntos] ¿Cuáles son las ventajas y desventajas del nivel 6 de la arquitectura de discos RAID respecto al nivel 5? Grafique la distribución de la información en los discos de ambos niveles.**

RAID 6

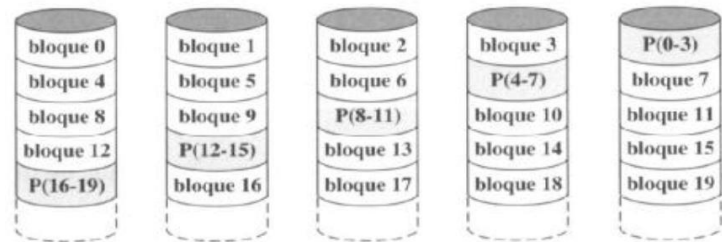
Ventajas:

- Proporciona una tolerancia a fallos mayor.
- Puede soportar la rotura de hasta 2 unidades simultáneamente.



Desventajas:

- Diseño del más complejo.
- Más costoso ya que se necesita un disco más (en nivel 5, n discos de datos +1 de redundancia; en nivel 6, n discos de datos +2 de redundancia).

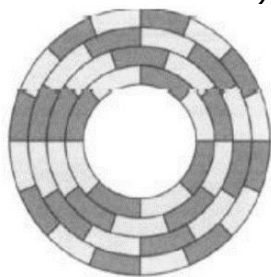
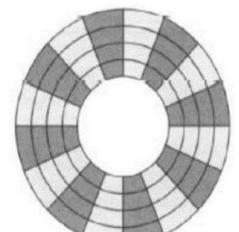


RAID 5

- **[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.**

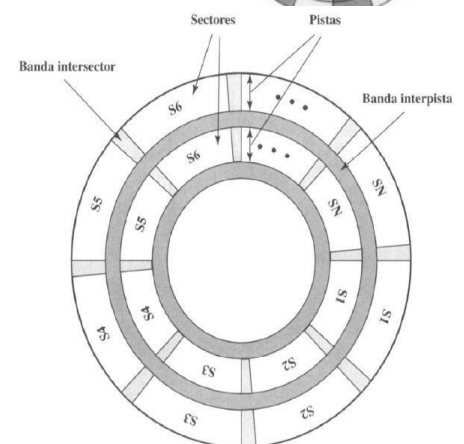
Ventajas:

Mayor capacidad de almacenamiento. En los discos tradicionales se tenían igual cantidad de bits en los sectores más alejados del centro y en los más cercanos. Esto era así debido a que se leía a una velocidad angular constante. En los discos multi zona esto cambia, y permite mayor densidad de bits en las zonas más alejadas. Por esto la capacidad de almacenamiento viene limitada por la máxima densidad de grabación que se puede llevar a cabo en la misma más interna (en los discos tradicionales).



Desventajas:

La mayor capacidad de almacenamiento viene acompañada de una circuitería más compleja y es más difícil situar a la cabeza en la zona que deseamos, ya que en el método tradicional esto se puede direccionar directamente con la pista y el sector.



- **[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.**

- Inmediato
 - mov ax,8h
- Registro-Registro
 - mov ax, bx
- Directo
 - mov ax, [200h]
- Indirecto
 - mov ax, [si]
 - mov ax, [si + 100h]
- Base/Relativo
 - mov ax, [bp]
 - mov ax, [bp + si + 8h]

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

Existen 2 mecanismos:

- Inhibición de interrupciones:

Deshabilita una interrupción mientras otra está siendo procesada. Si una interrupción ocurre en ese tiempo, queda pendiente para que el procesador la chequee. Entonces cuando un programa se ejecuta y aparece una interrupción, se deshabilitan las demás. Cuando la rutina de atención de interrupciones (RAI) se termina, las interrupciones se habilitan antes de ejecutar el programa y el procesador chequea si hubo otras interrupciones adicionales.

- Ventaja: El manejo es simple porque se habilitan/deshabilitan las interrupciones en un estricto orden secuencial.
- Desventaja: No toma en cuenta la prioridad relativa o las que son críticas.

- Manejo de prioridades

Permite definir prioridades a las interrupciones basándose en una tabla de prioridades

- Ventaja: Soluciona el problema con el otro mecanismo de que una interrupción más crítica debe ser llamada antes que las otras.

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

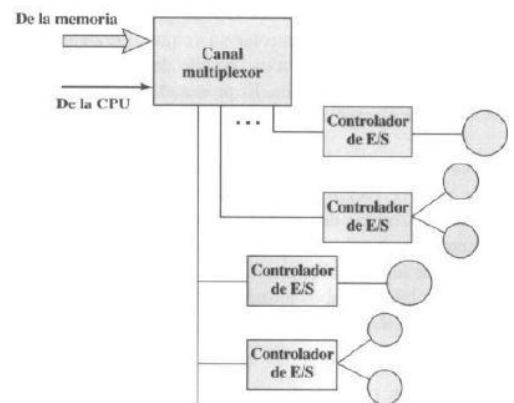
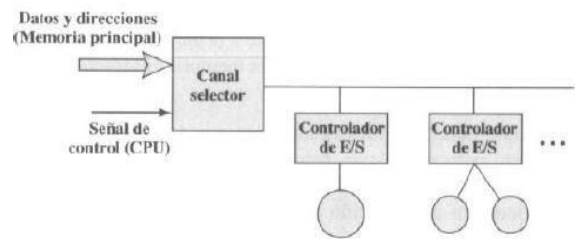
- Mayor velocidad de acceso a la información.
- Ruido (no hay).
- Calor (menos).
- Consumo de energía menor.
- Resistente a golpes.
- Seguridad al borrar datos.
- Rendimiento determinístico.

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

El canal de E/S representa una aplicación del concepto DMA. Un canal de E/S puede ejecutar instrucciones de E/S (propias), lo que le confiere un control completo sobre las operaciones de E/S. En un computador de tales dispositivos, la CPU no ejecuta instrucciones de E/S. Dichas instrucciones se almacenan en memoria principal para ser ejecutadas por un procesador de uso específico contenido en el propio canal de E/S.

Son comunes dos tipos

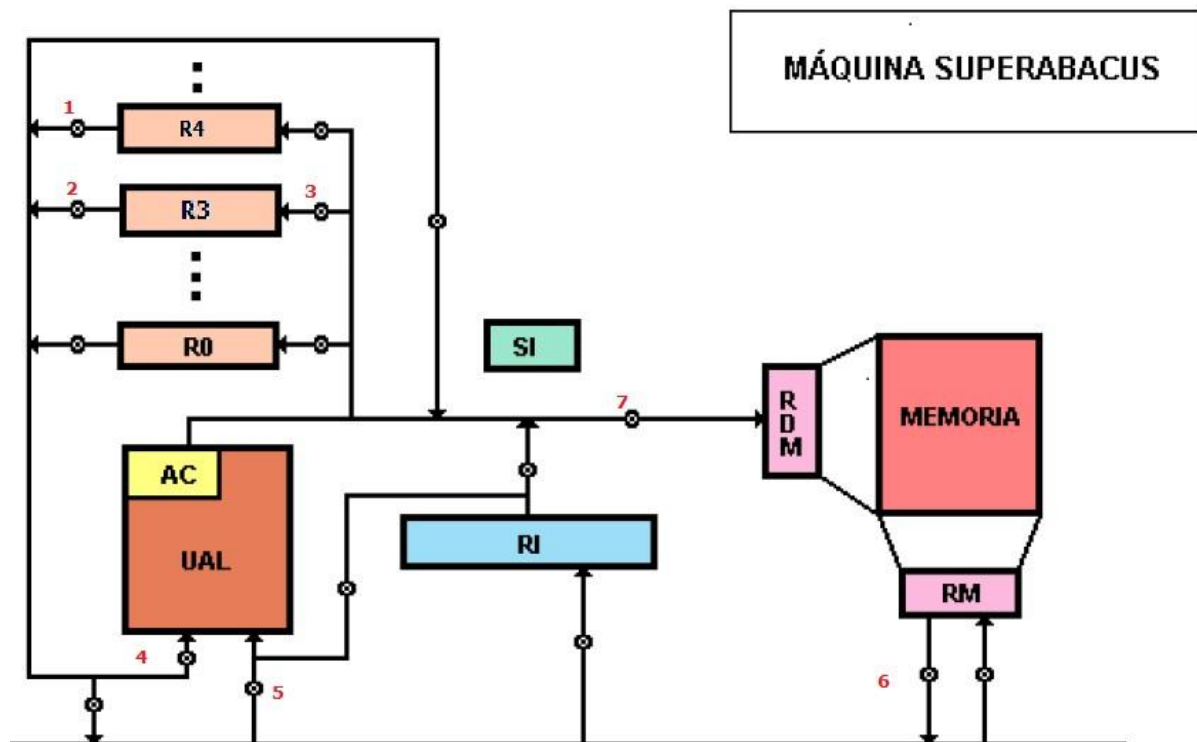
- Un canal selector
 - o Que controla varios dispositivos de velocidad elevada. El canal selecciona un dispositivo y efectúa la transferencia de datos. Cada dispositivo, o pequeño grupo de dispositivos, es manejado por un controlador.
- Canal multiplexor
 - o Puede manejar E/S de varios dispositivos al mismo tiempo. Para dispositivos de velocidad reducida.



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

- Decodificación de ordenes
 - o El módulo de E/S acepta órdenes del procesador. Estas órdenes generalmente se envían utilizando líneas del bus de datos.
- Información de estado
 - o Puesto que los periféricos son lentos, es importante conocer el estado del módulo de E/S. También puede haber señales para informar ciertas situaciones de error.
- Detección de errores
 - o El módulo de E/S es responsable de detectar errores e informar al procesador. Una clase de errores son los defectos mecánicos y eléctricos en funcionamiento del dispositivo. Otra clase son los cambios accidentales en los bits al transmitirse desde el dispositivo al módulo de E/S.

- **[1,5 puntos] Indique cuales son las microinstrucciones necesarias para ejecutar la instrucción SUMAR 3,4 en una maquina SuperAbacus, siendo 3 y 4 registros de uso general. Se pide además graficar en el esquema, el flujo de apertura de compuertas usadas en la fase de ejecución de dicha instrucción.**



AC	←	(R4)	1,4
AC	←	(AC)+R3	2,4
R3	←	(AC)	3

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

Método de acceso directo (discos magnéticos)

- Dirección única para bloques o registros basada en su posición física.
- Tiempo de acceso variable.

Método de acceso Aleatorio (Memoria principal, algunas memorias caché)

- Cada oposición direccionable de memoria tiene un mecanismo de direccionamiento cableado físicamente.
- Tiempo de acceso constante, independiente de la secuencia de accesos anteriores.

• **[1,5 puntos] ¿Qué es la codificación 8-14(EFM) y para que se usa?**

El protocolo 8-14 es una forma de codificar datos binarios, utilizada en medios ópticos. Se utiliza porque el sistema no es capaz de detectar dos 1 seguidos, entonces se intercalan 0. Para aplicarlo, se usa una tabla de doble entrada, donde a cada combinación de 8 bits le corresponde una de 14 bits sin unos contiguos.

• **[1 punto] ¿Qué ventajas presenta el modo de direccionamiento por desplazamiento (relativo al PC/referencia al programa) frente al direccionamiento directo?**

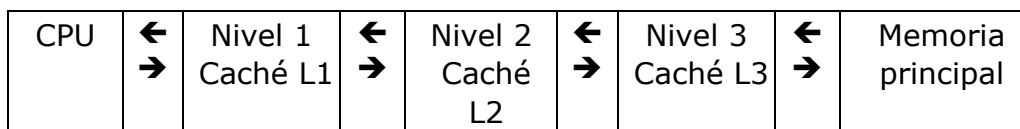
La ventaja del direccionamiento por referencia al programa y las demás formas de direccionamiento relativo es que permiten la reubicación de los programas en la memoria. En el caso particular del direccionamiento por referencia al programa, no

hay ningún cambio al reubicarlo ya que ese direccionamiento utiliza la dirección de la instrucción como referencia.

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

Su objetivo es lograr que la velocidad de la memoria sea lo más rápida posible. La caché contiene una copia de partes de memoria principal. Cuando el procesador intenta leer una palabra de memoria, se hace una comprobación para determinar si la palabra está en el caché. Si es así, se entrega dicha palabra. Si no, un bloque de memoria principal se transfiere a la caché y después la palabra es entregada al procesador.

Debido al fenómeno de localidad cuando un bloque de datos es capturado por la caché, es probable que se hagan referencias futuras a la misma posición de memoria o a otras palabras del mismo bloque.



- **[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 (en un sistema de cabeza móvil): El tiempo que tarda la cabeza en posicionarse en la pista.
- Latencia rotacional: Una vez en la pistas, es el tiempo que tarda el disco en girar hasta el sector apropiado.
- Tiempo de acceso: La suma del tiempo de seek y la latencia rotacional.
- Tiempo de transferencia de datos: Lo que se tarda en hacer la operación de lectura/escritura.

$$T_{total} = T_{seek} + \frac{1}{2}R + \frac{b}{RN}$$

R = Velocidad de rotación en revoluciones por segundo.

b = Bytes a transferir.

N = Bytes por pista

- **[1,5 puntos] ¿Cómo funciona el mecanismo de inhibición de interrupciones y para que se usa? ¿Qué desventaja tiene?**

Deshabilita una interrupción mientras otra interrupción está siendo procesada. Si una interrupción ocurre en este tiempo, queda pendiente para que el procesador la chequee. Entonces cuando un programa se ejecuta y aparece una interrupción, se deshabilitan las demás. Cuando la rutina de atención de Interrupciones (RAI) se termina, las interrupciones se habilitan antes de ejecutar el programa y el procesador chequea si hubo otras interrupciones adicionales.

- Ventaja: el manejo es simple porque se habilitan/deshabilitan las interrupciones en un estricto orden secuencial
- Desventaja: no toma en cuenta la prioridad relativa o las que son críticas.

- **[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.**

Valores no-numéricos: Denominados NaN (Not a number). Se identifican por un exponente con todos sus valores en 1, y un significando distinto de cero. Existen dos tipos de QNaN (Quiet NaN) y SNaN (Signalling NaN), que se distinguen dependiendo del valor 0/1 del bit más significativo del de la mantisa. QNaN tiene el primer bit en 1, y significa "Indeterminado". Resultado de todas aquellas operaciones aritméticas con resultados matemáticamente no definidos. SNaN tiene el primer bit en 0, y significa "operación no valida". Es la ejecución de una operación inválida. Ejemplos:

Operación	Resultado
Cualquier operación contra un NaN	NaN
$+0 / +0$	NaN
Infinito - Infinito	NaN
\pm Infinito / Infinito \pm	NaN
\pm Infinito X 0	NaN

	Signo	Exponente en exceso	Mantisa
QNaN	0/1	11111111	1 0101010101001010101101
SNaN	0/1	11111111	0 1010101101101010101010

- **[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.**
 - Para liberar al procesador del trabajo de E/S.
 - Para adaptar las velocidades de trabajo entre el CPU y los dispositivos.
 - Para proporcionar una interface estándar contra dispositivos disimiles.
- **[1 punto] Enuncie al menos 4 características de la arquitectura de procesadores CISC.**
 - Muchas instrucciones y muy complejas.
 - Muchos formatos de instrucciones.
 - Muchas instrucciones para acceder a operandos en memoria.
 - Muchos modos de direccionamiento.
 - Muchos tipos.
 - Pocos registros.
- **[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?**

Ventajas de administración de memoria paginada frente a otros mecanismos más sencillos:

- Minimiza la fragmentación interna (no del todo ya que se calcula que se pierde media página por programa).
- Permite que los bloques de memoria (frames) puedan almacenarse sin estar necesariamente contiguos. Esto evita tener que reordenar la memoria (compactamiento).

Desventaja de la administración de memoria paginada frente a la administración de memoria paginada por demanda.

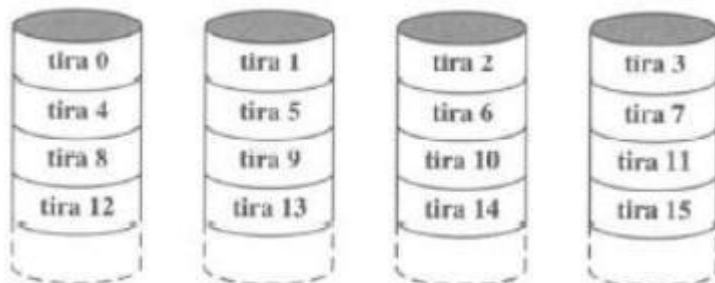
- Requiere que todas las páginas estén cargadas en la memoria.
- No permite ejecutar programas que no entran completamente en la memoria.

• **[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.**

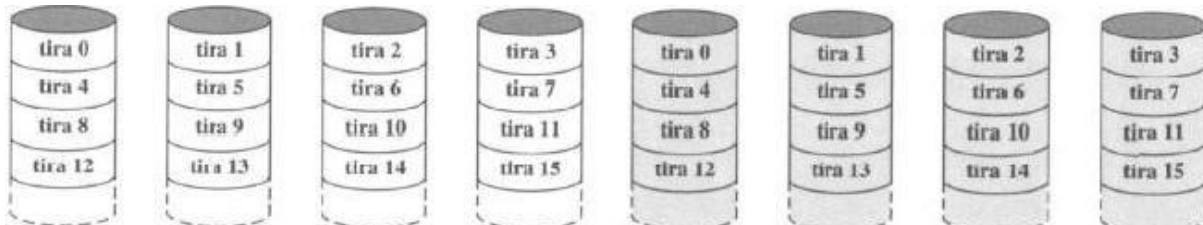
Ventajas

- Una petición de lectura puede ser servida por cualquiera de los discos que contienen los datos pedidos.
- La escritura se hace en forma independiente en cada disco.
- Alta disponibilidad de datos.
- La recuperación tras un fallo es muy sencilla, ya que están todos los datos por duplicado, lo que ofrece un 100% de redundancia.

RAID NIVEL 0



RAID NIVEL 1



• **[1 punto] Explique claramente qué es el fenómeno de "thrashing" y qué lo puede originar.**

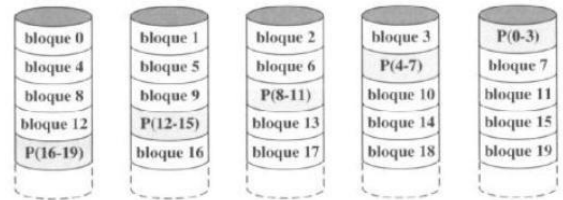
Si la CPU se ocupa de demasiados programas al mismo tiempo, como puede ocurrir cuando se usa paginación por demanda en un sistema con insuficiente memoria física (o algoritmos de juicio deficientes), no es raro que se carguen páginas en la memoria, se las suplante enseguida y luego se las vuelva a cargar, y así sucesivamente. A esto se lo llama thrashing y provoca un gran deterioro en la performance.

- **[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.**

RAID NIVEL 4

Ventajas

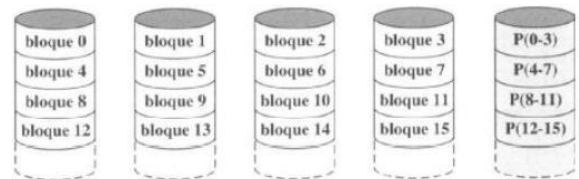
- Distribuye las tiras de paridad a lo largo de todos los discos. Lo cual evita un potencial cuello de botella de E/S encontrado en el RAID 4.



RAID NIVEL 5

Desventaja

- Controlador sumamente más complejo que el de su nivel inferior.



- **[1,5 pts] ¿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.**

Para los infinitos se ha convenido que cuando todos los bits del exponente están a 1 y todos los de la mantisa en 0, el valor es +/- infinito (según el bit de signo).

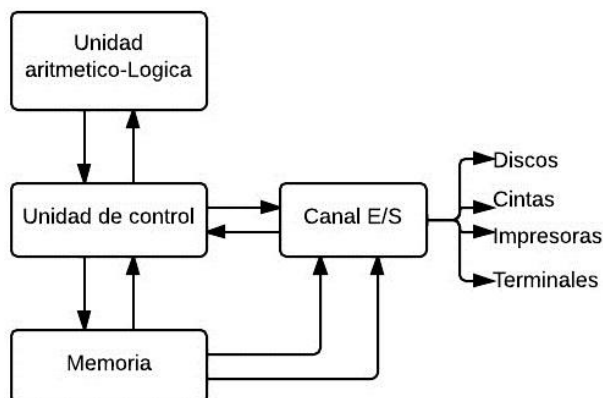
- **[1,5 puntos] Hacer un cuadro comparativo de los componentes de almacenamiento externos, dar características, ventajas y desventajas de c/u**

Nombre	Características	Ventajas	Desventajas
Cintas magnéticas	-Acceso secuencial. - Está compuesta por 9 pistas, 8 bits de datos y 1 de paridad. - Generalmente usado para backup.	-Vida útil. -Capacidad de almacenamiento. -Bajo costo.	-Muy lento. -Baja tasa de transferencia de datos.
Discos magnéticos	-Acceso directo basado en su posición física. -Hay dos maneras en las que se disponen la información, la tradicional o la multizona.	-Mejor tasa de transferencia de datos que cintas magnéticas. -Mucha capacidad de almacenamiento (menor que cintas pero mayor que SSD).	-Genera mucho calor. -Ruidoso. -Fragmentación de datos en discos muy usados, esto puede generar un acceso casi aleatorio.

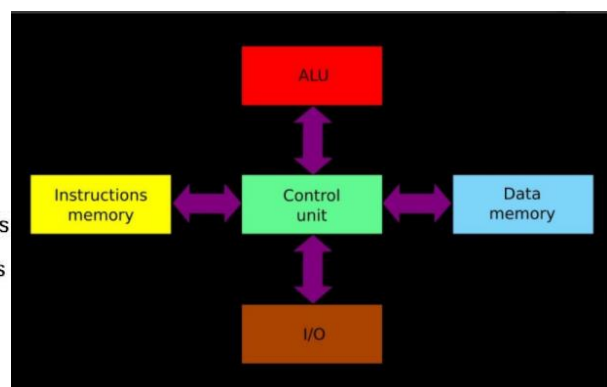
Medios ópticos	-Existe diferentes tipos (CD-Audio, CDROM,...,DVD, Blu-ray,...)	-Buena portabilidad. - Buena transferencia de datos.	-Poca vida útil. -Poca capacidad. -
Disco de estado solido	-Posee un controlador, un buffer, memoria caché y un condensador.	-Silencioso -Rápida transferencia de datos. -No produce calor. -No tiene elementos mecánicos. -Mejor velocidad de acceso a la información -Consumo mínimo de energía. -Peso/Tamaño reducido	-Mayo precio por byte. -Menor capacidad que los discos magnéticos. -Menor vida útil que discos y cintas magnéticas. -No hay recuperación de la información ante fallos.

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

- Ventajas:
 - La división de la memoria, en una memoria de instrucciones y una memoria de datos
 - El procesador puede acceder a ambas memorias simultáneamente
 - Cada memoria y cada conexión de estas puede tomar distintas características. Tamaño de la memoria o la tecnología para su implementación
 - Mapa de direcciones y mapa de datos separados
- Desventajas:
 - La división de memorias solo funciona mejor en el caso particular, donde las frecuencias de lectura de instrucciones y de datos es aproximadamente la misma



Von Neuman



Harvard

- **[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**

La instrucción es ejecutada sólo si el estado actual del flag del código de condición del procesador satisface la condición especificada en los bits b 31 -b 28 de la instrucción. Por lo tanto, las instrucciones cuya condición no se ve satisfecha en el flag de código de condición del procesador no se ejecutan. Una de las condiciones se utiliza para indicar que la instrucción siempre se ejecuta.

Esta característica elimina la necesidad de utilizar muchas bifurcaciones. El costo en tiempo de no ejecutar una instrucción condicional es frecuentemente menor que el uso de una bifurcación o llamado a una subrutina que, de otra manera, sería necesaria.

Para que una instrucción sea ejecutada condicionalmente se le agrega el sufijo con la condición apropiada. Por ejemplo, una instrucción de suma tiene la siguiente forma:

```
ADD r0, r1, r2
```

y para ejecutarla sólo si el flag cero está seteado:

```
ADDEQ r0, r1, r2
```

- **[1,5 puntos] Explique claramente como funciona el Linking estático. Ejemplifique y grafique dicho funcionamiento**

Cada módulo objeto compilado o ensamblado es creado con referencias relativos al inicio del módulo. Donde se combinan todos los módulos objeto en un único load module reubicable con todas las referencias relativas al load module.

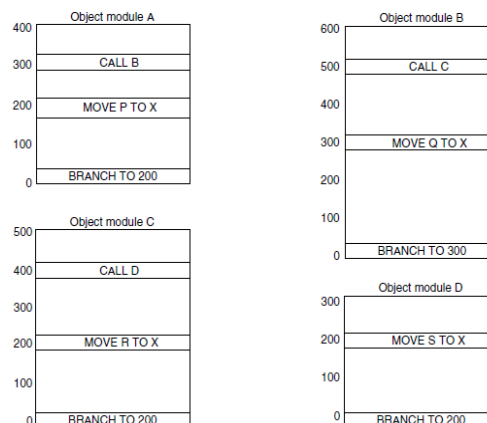


Figure 7-13. Each module has its own address space, starting at 0.

Module	Length	Starting address
A	400	100
B	600	500
C	500	1100
D	300	1600

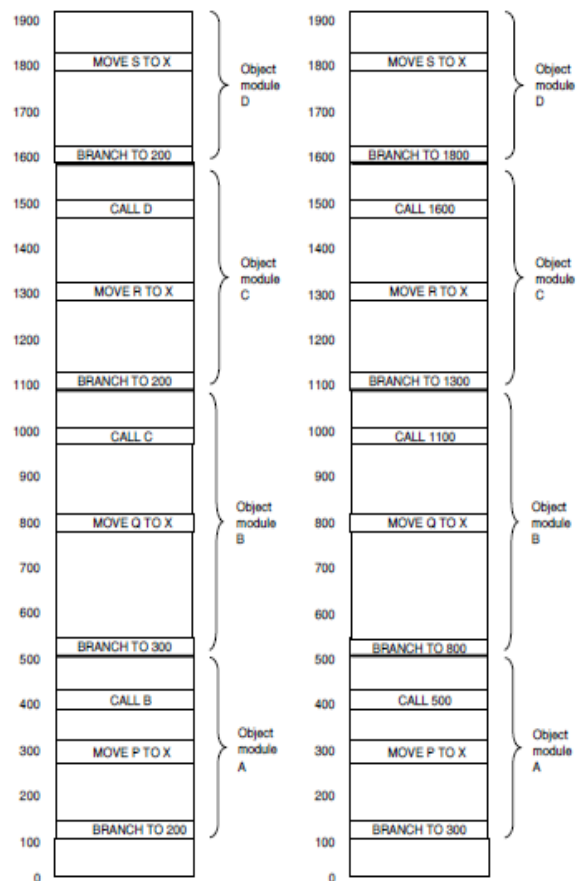


Figure 7-14. (a) The object modules of Fig. 7-13 after being positioned in the binary image but before being relocated and linked. (b) The same object modules after linking and after relocation has been performed.

- **[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?**

Cintas magnéticas son usadas actualmente para hacer backup y de archivo de información.

Ventajas:

1. Vida útil, pueden llegar a durar hasta 30 años
2. Posee grandes tamaños de almacenamientos de hasta PB
3. Bajo costo por byte: Con respecto a los demás medios de almacenamientos, estos se pueden producir a menor costo dado que el material usado es más barato y no requiere de componentes eléctricos, si lo comparamos con los medios ópticos, estos tienen poco espacio de almacenamiento

Desventajas:

- a Muy lento: Acceso secuencial a la información: si estoy en el registro 1 y quiero llegar al N tengo que "leer" los N-1 del medio. Si quiero leer un registro anterior tengo que rebobinar y volver a buscar el registro
- b Baja tasa de transferencia de datos. Por tener un grabado secuencial

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

1. Identificación: nombre del módulo, longitudes de las partes del módulo
2. Tabla de punto de entrada: lista de símbolos que pueden ser referenciados desde otros módulos
3. Tabla de referencias externas: lista de símbolos usados en el módulo, pero definidos fuera de él y sus referencias en el código
4. Código ensamblado y constantes
5. Diccionario de reubicabilidad: lista de direcciones a ser reubicadas
6. Fin de módulo

End of module
Relocation dictionary
Machine instructions and constants
External reference table
Entry point table
Identification

- **[1 punto] Indique como se puede clasificar el repertorio de instrucciones de una arquitectura de computadores de acuerdo con el número de direcciones. Ejemplifique y/o grafique cada uno.**

El conjunto de

1. 0 direcciones (Stack)

Ejemplo: add

$TOS \leftarrow TOS + Next$

2. 1 dirección (Acumulador)

Ejemplo: add A

$AC \leftarrow AC + Mem[A]$

3. 2 direcciones (Reg-Mem/Reg-Reg/Mem-Mem)

Ejemplo: add R1, A

$R1 \leftarrow R1 + Mem[A]$

4. 3 direcciones (Reg/Mem)

Ejemplo: add R1, R2, R3

$R1 \leftarrow R2 + R3$

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

Nombre	Nombre Alternativo	Ejemplos
Registro a registro	Registro directo	mov r0, r1
Absoluto	Directo	ldr r0, mem
Literal	Inmediato	mov r0, #15 add r1, r2, #12
Indexado, base	Registro indirecto	ldr r0, [r1]
Pre-Indexado, base con desplazamiento	Registro indirecto con offset	ldr r0, [r1, #4]
Pre-indexado, autoindexado	Registro indirecto con pre-incremento	ldr r0, [r1, #4]!
Post-indexado, autoindexado	Registro indirecto con post-incremento	ldr r0, [r1], #4
Doble registro indirecto	Registro indirecto indexado	ldr r0, [r1, r2]
Doble registro indirecto escalado	Registro indirecto indexado escalado	ldr r0, [r1, r2, lsl #2]
Relativo al PC		ldr r0, [PC, #offset]

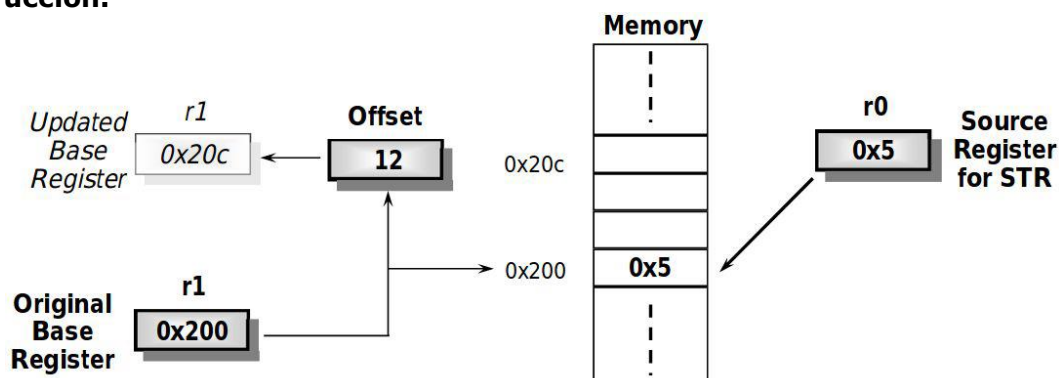
- **[1,5 puntos] Indique claramente que es Linking dinámico en tiempo de ejecución y cuáles son las diferencias frente al Linking dinámico**

1. Se pospone el linkeo hasta el tiempo de ejecución
2. Se mantienen las referencias a módulos externos en el programa cargado
3. Cuando efectivamente se invoca al módulo externo, el sistema operativo lo busca, lo carga y linkea al módulo llamador.

Ventajas

- i No ocupo memoria hasta que la necesito (ej. Bibliotecas DLL de Windows)

- **[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.**



La dirección efectiva del operando es el contenido de Rn. El desplazamiento se agrega a esta dirección y el resultado se escribe de nuevo en Rn.

`[Rn], #offset`

$DE = [Rn]$

$Rn \leftarrow [Rn] + \text{offset}$

`[Rn], ±Rm, shift`

$DE = [Rn]$

$Rn \leftarrow [Rn] \pm [Rm] \text{ shifteado}$

Primero guardo después actualizo!

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

ARM no tiene instrucciones de shift. En su lugar tiene un barrel shifter que provee un mecanismo que lleva a cabo shifts como parte de otras instrucciones.

Cuando se especifica que el segundo operando es un registro shifteado, la operación del Barrel Shifter es controlada por el campo Shift en la instrucción.

Este campo indica el tipo de shift a realizar. La cantidad de bits a shiftear puede estar contenida en un campo inmediato o en el byte inferior de otro registro (que no sea el R15)

1. Shift a Izquierda (LSL): Shift a la izquierda según la cantidad especificada (multiplica por potencias de 2). Por ejemplo:
`LSL # 5 @ multiplica por 32`
2. Shift Lógico a Derecha (LSR): Shift a la derecha según la cantidad especificada (divide por potencia de 2). Por ejemplo:
`LSR # 5 @ divide por 32`
3. Shift Aritmético a Derecha (ASR): Shift a la derecha según la cantidad especificada (divide por potencia de 2) preservando el bit de signo. Por ejemplo:
`ASR # 5 @ divide por 32`

Preguntas con ejercicios de codificación

- [2 puntos] 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_PrStr, 0x69
.equ SWI_Exit, 0x11

.data
first_string:
.asciz "Hola\n"
second_string:
.asciz "Chau\n"
third_string:
.asciz "como estas?\n"

.text
.global _start
_start:
    ldr r3, =first_string
    bl print_r3
    ldr r3, =second_string
    bl print_r3
    ldr r3, =third_string
    bl print_r3
    b    fin

print_r3:
    stmfd sp!, {r0,r1,lr}
    mov r0, #1
    mov r1, r3
    swi SWI_Print_String
    ldmfd sp!, {r0,r1,pc}

fin:
    swi SWI_Exit
.end
```

- [2 puntos] 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(restar) 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 suma
        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

```

- [2 puntos] 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 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,r2}
    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
```

- [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

```

.equ SWI_Open_File, 0x66
.equ SWI_Read_Int, 0x6C
.equ SWI_Print_Int, 0x6B
.equ SWI_Close_File, 0x68
.equ SWI_Exit, 0x11
.equ SWI_Print_Char, 0x00
.equ SWI_Print_Str, 0x69
.equ Stdout, 1
.data
filename:
.asciz "enteros.txt"
eol:
.asciz "\n"
.align
InFileHandle:
.word 0
.text
.global _start
_start: @abrir archivo
    ldr r0, =filename           @ nombre de archivo de entrada
    mov r1, #0                  @ modo: entrada
    swi SWI_Open_File           @ abre archivo
    bcs InFileError             @ chequear si hubo error
    ldr r1,=InFileHandle        @ cargar dirección donde almacenar el handler
    str r0, [r1]                @ almacenar handler
    mov r4, #1                  @acumulador de productoria
read_loop: @leer entero de archivo
    ldr r0, =InFileHandle
    ldr r0, [r0]
    swi SWI_Read_Int
    bcs EofReached
    mov r2, r0                  @ el entero está ahora en r0
    @Compruebo si es positivo
    mov r3, #0
    subs r3, r3, r2              @Si la resta es negativa, r2>0
    bmi productoria
    b read_loop
productoria:
    mul r4, r4, r2
    b read_loop
InFileError:
EofReached:
    ldr r0, =Stdout
    mov r1, r4
    swi SWI_Print_Int
    swi SWI_Exit
.end

```