

BALOTARIO ARQUI

1.Describa en que consiste el DataPath, los elementos que lo conforman, la primera parte y la segunda parte del Datapath. ¿Como se lee y escribe un registro?

- Es la componente del CPU que ejecuta las operaciones aritméticas y lógicas.
- Los elementos que lo conforman son: Banco de Registros, ALUs, multiplexores, memorias(de instrucciones y de datos),sumadores,etc.
- La primera parte comprende un Ciclo de fetch donde se lee una instrucción y se incrementa el PC.
- La segunda parte es la implementacion de la instrucción mediante instrucciones aritmeticas y logicas como ademas de instrucciones de carga y almacenaje.

2.En instrucciones de formato R indique como es el procedimiento de ejecucion de estas instrucciones.

- ☐ Lee dos registros que son operandos en la instrucción
- ☐ Ejecuta operación aritmética/lógica
- ☐ Escribe resultado en el registro

3. Explique el principio de localidad de la memoria y la jerarquia de memoria. Porque es importante la jerarquia de la memoria?. Que es un bloque?. Cuando se produce un acierto y un fallo de datos?. Como se corrige un fallo?.

- El principio de localidad de memoria establece que si un programa accede a la memoria es muy posible que se mueva dentro de un pequeño espacio compacto de memoria, y que este espacio sea usado pronto.
- La jerarquia de memoria establece que la memoria consiste en multiples niveles de memoria con diferentes velocidades y tamaños, cuando el tamaño de la memoria accedida es pequeño la velocidad de accesibilidad es alta, mientras que cuando es grande en tamaño, es baja en velocidad.
- La jerarquia de memoria es importante debido a que podemos ubicar en memorias pequeñas y rapidas toda la información que es constantemente usada y actualizada por el CPU,mientras que la información que se mantiene estacionaria es colocada en un nivel de memoria inferior donde la memoria es grande y lenta.
- Un bloque es la unidad minima de informacion que se encuentra en la memoria.
- Los aciertos se producen cuando la información buscada por el CPU esta en el nivel superior, un fallo es cuando no se encuentran en el nivel superior, pasando a buscar en el nivel inferior.
- Se corrige un fallo remplazando la informacion necesaria del nivel inferior al superior.

4.Explique “dirección virtual” dentro de la memoria virtual. Use la analogía de la biblioteca. Explique como funciona el TLB.

- En la memoria virtual la CPU produce una dirección virtual que es traducida por hardware y software a una dirección física que se usa para acceder a la memoria principal.
- Esta traducción se llama mapeo de direcciones o traducción de direcciones.
- Hoy en día, los niveles de la jerarquía de la memoria controlados por la memoria virtual corresponden a las DRAMs y a los discos magnéticos.
- Según la analogía con la BIBLIOTECA, podemos pensar que una dirección virtual se corresponde con el título del libro; y que una dirección física se corresponde con su ubicación en la biblioteca donde se encuentra el libro, que le fue asignada según el código de identificación de la biblioteca.
- TLB: búfer de traducción lateral, actúa como caché para las entradas que mapean solo a páginas físicas, contiene un subconjunto de los mapeos de página virtual a página física que están en la table de páginas, como es un caché tiene campo de etiqueta.

5.Explique cada uno de los métodos utilizados en la comunicación con el procesador.

- **ENCUESTA(polling).** Comprueba periódicamente el estado de un dispositivo de E/S para determinar la necesidad de dar servicio al dispositivo.
- **E/S dirigida por interrupciones.** Emplea interrupciones para indicar al procesador que un dispositivo de E/S necesita atención.
- **DMA(Acceso directo a memoria).** Mecanismo que proporciona a un controlador de dispositivo la capacidad de transferir datos directamente hacia o desde la memoria sin la participación del procesador.

6.Mencione al menos 3 lenguajes de programación que se utilizan para programar un GPU. Puede un GPU remplazar a un CPU?

- DirectX, OpenGL
- C para gráficos (Cg), High Level Shader Language(HLSL)
- Compute Unified Device Architecture (CUDA) de NVIDIA

No, debido a que las GPUs son aceleradores que complementan a la CPU, no realizan todas las funciones de la CPU.

7.Describa brevemente la historia del procesador MIPS y describa los circuitos combinatorios.

HISTORIA:

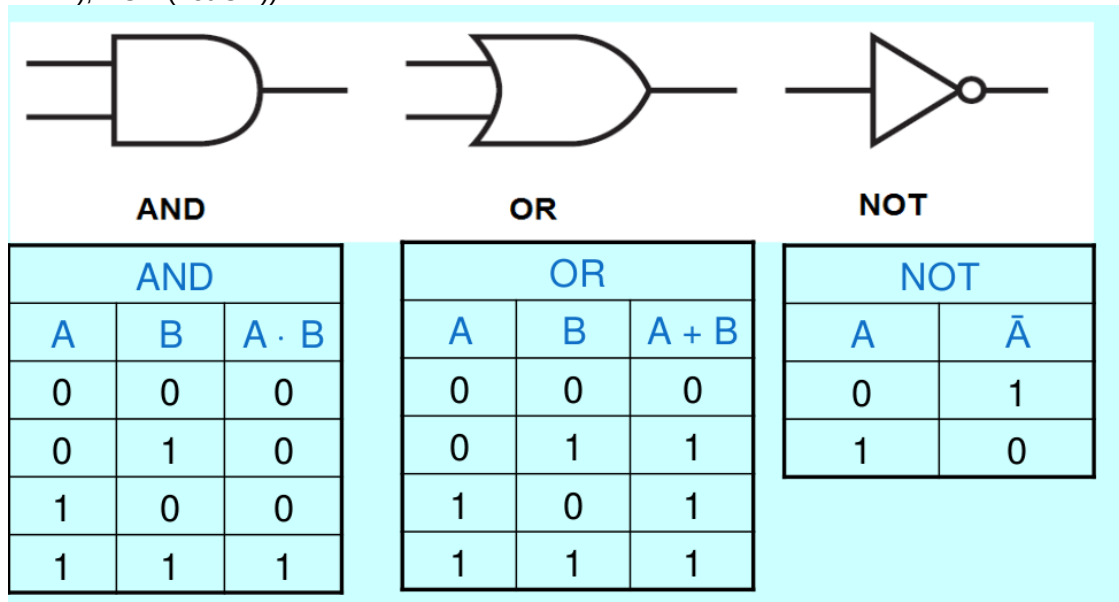
- John L. Hennessy en 1981, comenzó a formular la creación del primer procesador MIPS, su idea consistía en mejorar el rendimiento de las CPU's mediante el uso de la segmentación, técnica que permitía la ejecución de una instrucción sin que una anterior haya terminado.
- En 1984 Hennessy funda la empresa MIPS Computer Systems, luego en 1985 la empresa presenta el R2000 (primer diseño) con su mejora R3000 en 1988, ambas CPU's de 32 bits.
- El R3000A, utilizado en la extremadamente exitosa Sony PlayStation, fue una versión acelerada hasta los 40 Mhz.
- En 1991 MIPS presentó su primer procesador de 64 bits, el R4000.
- En 1992 SGI compra MIPS Computer Systems y la nombra MIPS Technologies.

- Entre sus versiones mas recientes tenemos a R16000 y R16000A.

CIRCUITOS COMBINATORIOS:

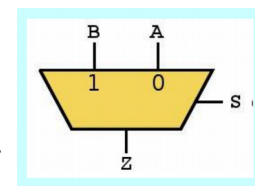
Un circuito combinatorio es un arreglo de compuertas lógicas, multiplexores, sumadores y otros componentes donde las salidas dependen exclusivamente de sus entradas.

- Compuertas logicas básicas** (existen tambien XOR (or exclusivo), EQV (equivalence), NAND (not AND), NOR (not OR)) :



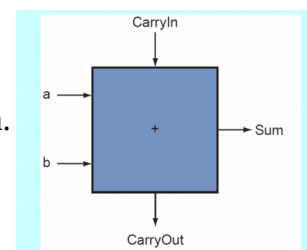
- Multiplexor:**

Se usa para elegir que entrada pasara a la salida.
Esto se realiza mediante la configuración de los bits de selección.



- Sumador Completo:**

- ☑ Entradas: dos números de 1 bit y un bit de carry de entrada.
- ☑ Salidas: la suma de 1 bit y un bit de carry de salida.



8. En que consiste la estrategia “Cache de mapeo directo”?

A cada dato se le asigna un lugar en el caché de acuerdo a su dirección en la memoria principal.

$$b = d \bmod n$$

b es el bloque que le corresponde en el caché

d es la dirección de bloque en el caché

n es el número de bloques del caché

Si n es 2 a la m, el caché se puede indexar con los m bits mas bajos de su dirección de bloque.

$d = a \cdot \text{div}(k)$

d es la dirección de bloque en el caché

a es la dirección del dato en memoria

k es el número de bytes por bloque

9. Describa los tipos de organización de la memoria.

En la organización ANCHO DE 1 PALABRA

Ancho del bus y de la memoria es de 1 palabra

El acceso a la memoria es secuencial

Castigo por fallo: $1 + 4 \times 15 + 4 \times 1 = 65$ ciclos de reloj del bus

Número de bytes transferidos por ciclo de reloj del bus por cada fallo: $(4 \times 4) / 65 = 0.25$

En la organización MEMORIA ANCHA

Ancho del bus y de la memoria es de 4 palabras

El acceso a la memoria es paralelo

Castigo por fallo: $1 + 1 \times 15 + 1 \times 1 = 17$ ciclos de reloj del bus

Número de bytes transferidos por ciclo de reloj del bus por cada fallo: $(4 \times 4) / 17 = 0.94$

En la organización INTERLEAVED(entrelazado):

Ancho del bus es de 1 palabra

Ancho de la memoria es de 4 palabras repartidas en 4 bancos de memoria independientes.

El acceso a la memoria es en paralelo.

La transferencia es secuencial

Castigo por fallo: $1 + 1 \times 15 + 1 \times 4 = 20$ ciclos de reloj del bus

Número de bytes transferidos por ciclo de reloj del bus por cada fallo: $(4 \times 4) / 20 = 0.8$

El castigo por fallo es aceptable.

Un bus y memoria de 4 palabras es más caro que un bus y 4 bancos de 1 palabra cada uno.

Cada banco puede escribir en forma independiente, cuadruplicando el ancho de banda al escribir y provocando menos detecciones(stalls) en un caché write-through.

10. Describa las características de la memoria virtual.

Es un mecanismo que permite que la memoria principal parezca más grande que su tamaño físico..

Permite ejecutar programas más grandes que la memoria física disponible.

Un bloque en memoria virtual se llama página

Un fallo en memoria virtual se llama falta de página o fallo de página

En memoria virtual la CPU produce una dirección virtual que es traducida por hardware y software a una dirección física que se usa para acceder a la memoria principal.

11. Describa los tipos de buses que existen en un sistema computador.

• **Buses Procesador-Memoria**

-Pequeña longitud(corta) y de alta frecuencia

-Diseñado para maximizar el ancho de banda Procesador-Memoria

• **Buses de E/S**

-Más largas, permite múltiples conexiones

-Especificado por normas(standards) para interoperabilidad

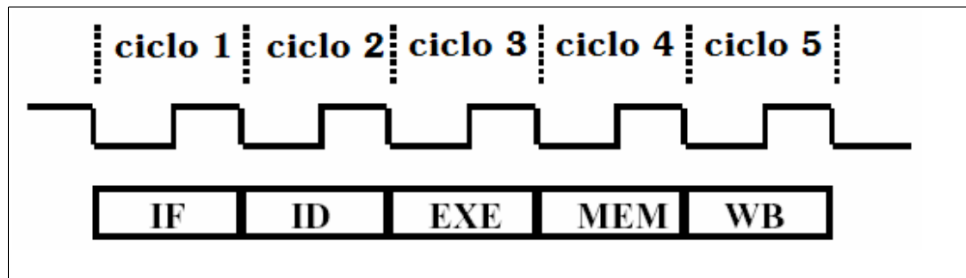
-Se conecta al bus Procesador-Memoria a través de un puente.

• **Buses gráficos**, son otros con características diferentes y que realizan funciones especiales.

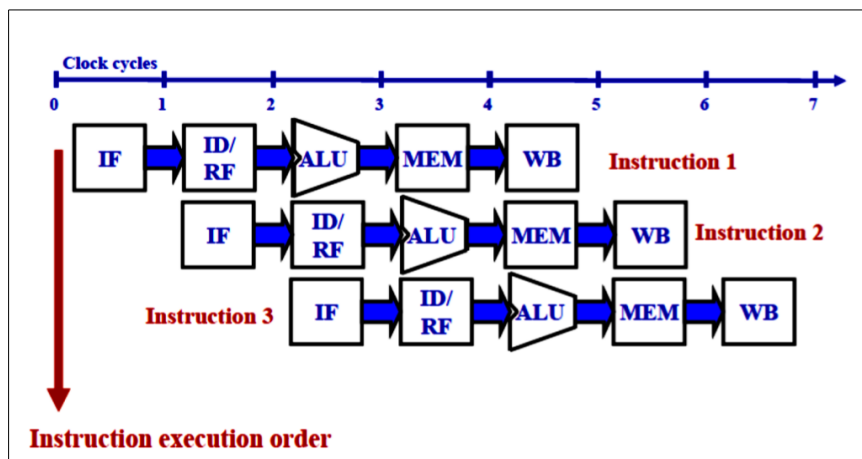
12.Describa el pipelining en MIPS.

Las instrucciones MIPS presentan cinco estados (cada estado toma un ciclo de reloj):

- IF: Búsqueda de instrucción
- ID: Búsqueda de registros y decodificación de instrucciones
- EXE: Ejecución o cálculo de dirección
- MEM: Acceso a la memoria de datos.
- WB: Escribir datos en el archivo de registros.



El **pipelining** Es una técnica de implementación por medio de la cual se puede traslapar la ejecución de instrucciones, en vez de realizarlas secuencialmente, esto se muestra en la siguiente figura:



Cada instrucción puede iniciar sin que la anterior termine, sin embargo, Hay situaciones en la segmentación cuando la siguiente instrucción no se puede ejecutar en el próximo ciclo de reloj, estas situaciones se llaman riesgos, entre las mas importantes mencionamos :

- **Riesgos estructurales;** El hardware no puede manejar la combinación de instrucciones que deben ejecutarse en el mismo ciclo. (suponiendo que se cuenta con una única memoria). Ejemplo: IF y MEM no pueden ejecutarse en el mismo ciclo ya que ambas necesitan accesar datos de la memoria.

| ciclo | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|--------|----|----|----|-----|-----|-----|-----|----|
| inst 1 | IF | ID | EX | MEM | WB | | | |
| inst 2 | | IF | ID | EX | MEM | WB | | |
| inst 3 | | | IF | ID | EX | MEM | WB | |
| inst 4 | | | | IF | ID | EX | MEM | WB |

↘ riesgo estructural

- **Riesgos de datos;** ocurren porque una instrucción depende del resultado de una instrucción previa en el segmento.

13. Defina el procesador superescalar, el procesador vectorial y el GPU.

Procesador superescalar

Es un desarrollo de la arquitectura con pipeline (segmentación).

Se generan nuevos peligros por dependencias.

Emplea una técnica avanzada de segmentación que le permite ejecutar más de una instrucción por ciclo de reloj. Un procesador (multinúcleo) superescalar es de tipo MIMD.

Procesador vectorial

Procesador capaz de ejecutar operaciones matemáticas sobre múltiples datos de forma simultánea, en contraste con los procesadores escalares, capaces de manejar sólo un dato cada vez, proporcionan operaciones de alto nivel que trabajan sobre vectores, segmenta las operaciones sobre los elementos de un vector, tanto la operación aritmética como los accesos a la memoria.

GPU

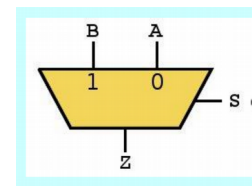
Es un co-procesador dedicado al procesamiento de gráficos u operaciones de coma flotante, para aligerar la carga del procesador central en aplicaciones como los videojuegos o aplicaciones 3D interactivas. De esta forma, mientras gran parte de lo relacionado con los gráficos se procesa en la GPU, la CPU puede dedicarse a otro tipo de cálculos (como la IA o los cálculos mecánicos en el caso de los videojuegos).

14. ¿Qué es un multiplexor? ¿Cómo se llama el dispositivo que puede memorizar? Explique

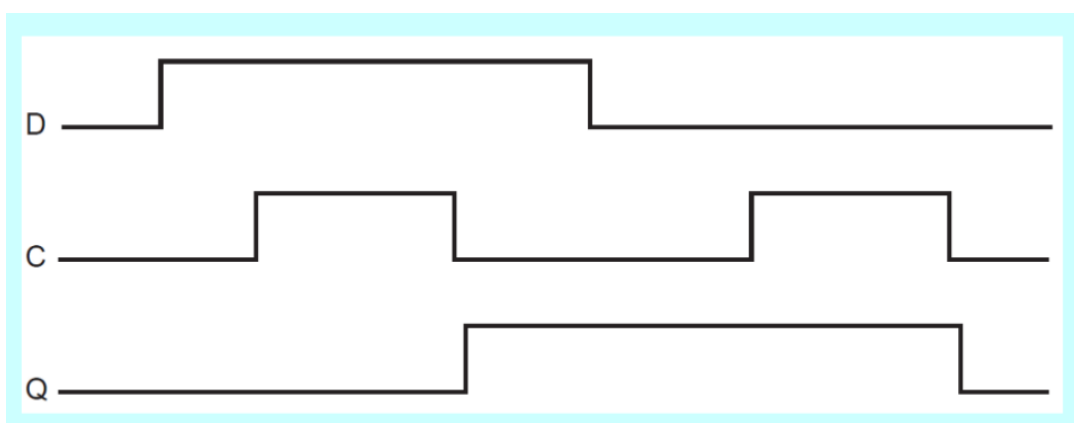
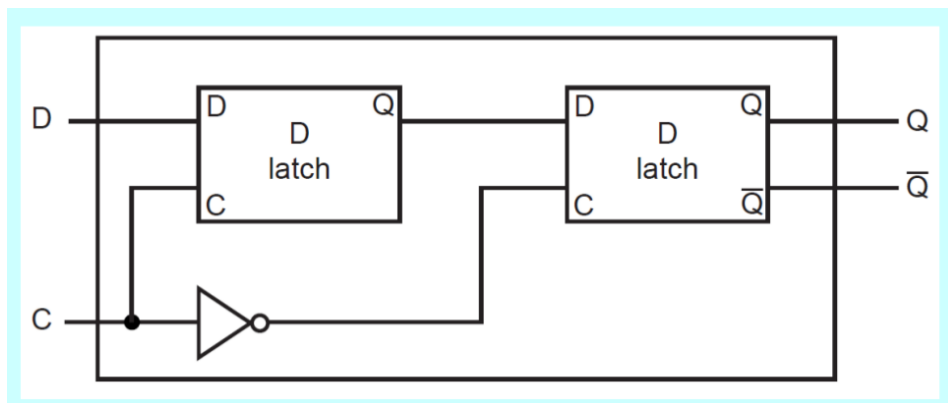
- **Multiplexor:**

Se usa para elegir que entrada pasara a la salida.

Esto se realiza mediante la configuración de los bits de selección.



- El dispositivo que puede memorizar se llama **Flip-flop D**. Su función es dejar pasar lo que entra por D, a la salida Q, después de un pulso del reloj (coincide con el flanco de bajada en C). Por lo que se permite memorizar la señal D, a pesar de que esta cambie. En la imagen, C es nuestro reloj.



15. Explique la ley de Amdahl. ¿Para que sirve?

Sirve para evaluar como cambia el rendimiento al mejorar solo una parte de la computadora.

Define el speedup global (aceleración) que se puede alcanzar al usar cierta mejora.

$\text{Speedup}_{\text{global}} = \text{Tiempo de ejecución sin la mejora} / \text{Tiempo de ejecución con la mejora}$

La ley dice que:

$\text{Speedup}_{\text{global}} = 1 / [(1 - \text{Fracción mejora}) + (\text{Fracción}_{\text{mejora}} / \text{Speedup}_{\text{mejora}})]$

concluyendo que:

$\text{Tiempo con mejora} = \text{Tiempo sin mejora} * [(1 - \text{Fracción mejora}) + (\text{Fracción}_{\text{mejora}} / \text{Speedup}_{\text{mejora}})]$