

ESTRUCTURA DE COMPUTADORES

Tema de Procesador (Enunciados de problemas)

PROBLEMAS

1 La figura 1 muestra la estructura de la CPU de un computador con palabras y direcciones de 32 bits, cuya unidad de control es microprogramada. El banco de registros (BR) contiene 25 registros generales (R1 a R25) y una puerta (A) de lectura/escritura. El registro de estado es de 6 bits. Ninguno de los registros de este computador permite autoincremento o autodecremento. La memoria es asíncrona, lo que se indica mediante la señal WAIT. Las operaciones que realiza la ALU aparecen en el recuadro inferior de la figura.

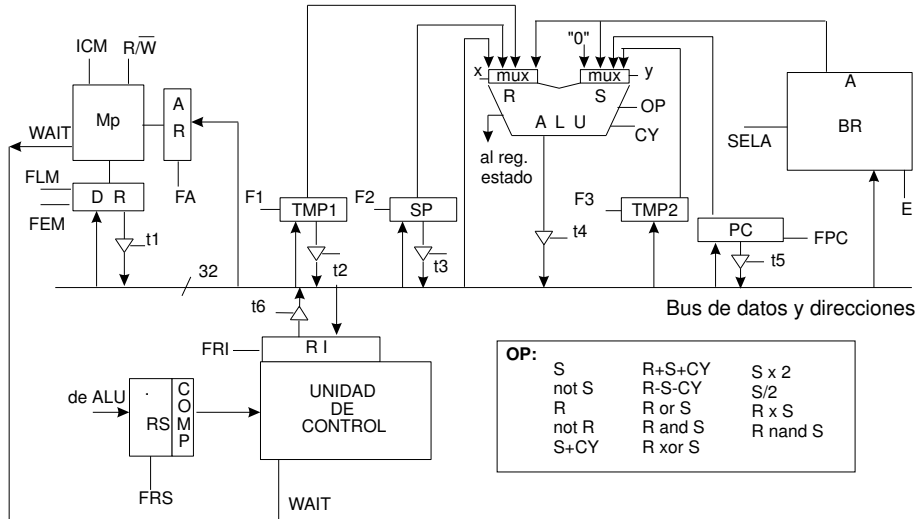


Figura 1 Estructura de la CPU y operaciones de la ALU

1. Realice, a nivel RT, el microprograma de la instrucción de dos palabras `ADD_INT .R1, .R2, /Dir` cuyo formato se muestra en la figura 2:

C.O.	Reg.	Reg.	No utilizado
12	5	5	10
Dir			
32			

Figura 2 Formato de la instrucción

Esta instrucción suma el contenido de dos registros del banco de registros e intercambia el resultado con el contenido de una dirección de memoria. El intercambio sólo se produce si el resultado de la suma es distinto de cero, en cuyo caso, se almacena en la dirección de memoria especificada en la segunda palabra de la instrucción y el contenido de ésta se carga en ambos registros.

2. De acuerdo con las características del computador representado en la figura 1 y, teniendo en cuenta, la siguiente información:

- El juego de instrucciones consta de 200 instrucciones.
- El tamaño medio de cada microprograma es de 30 palabras de control.
- Se reservan 144 posiciones para la microprogramación de microsubrutinas.

Diseñe el formato de las microinstrucciones y calcule el tamaño mínimo de la memoria de control de este computador.

2 En la figura 1 se muestra esquemáticamente la estructura de la CPU de un computador con palabras y direcciones de 32 bits y unidad de control microprogramada con pipeline RC-RS (Registro de Control-Registro de Estado), que dispone de los siguientes registros:

- *TMP1 y TMP2: Registros temporales, transparentes al usuario*
- *BR: banco de registros con 12 registros generales (R1 a R12) y una puerta de lectura/escritura (A)*
- *AR: Registro de direcciones de memoria*
- *DR: Registro de datos de memoria*
- *RI: Registro de instrucción*
- *PC: Contador de programa*
- *SP: Puntero de pila, que apunta a la última dirección ocupada de la pila*
- *RS: Registro de estado, de 8 bits*

Ninguno de los registros permite autoincremento o autodecremento. Las operaciones que es capaz de realizar la ALU aparecen en el recuadro inferior de la figura 1. En este computador la memoria funciona de forma asíncrona dependiendo de la activación de la señal WAIT y la pila crece hacia direcciones de memoria crecientes.

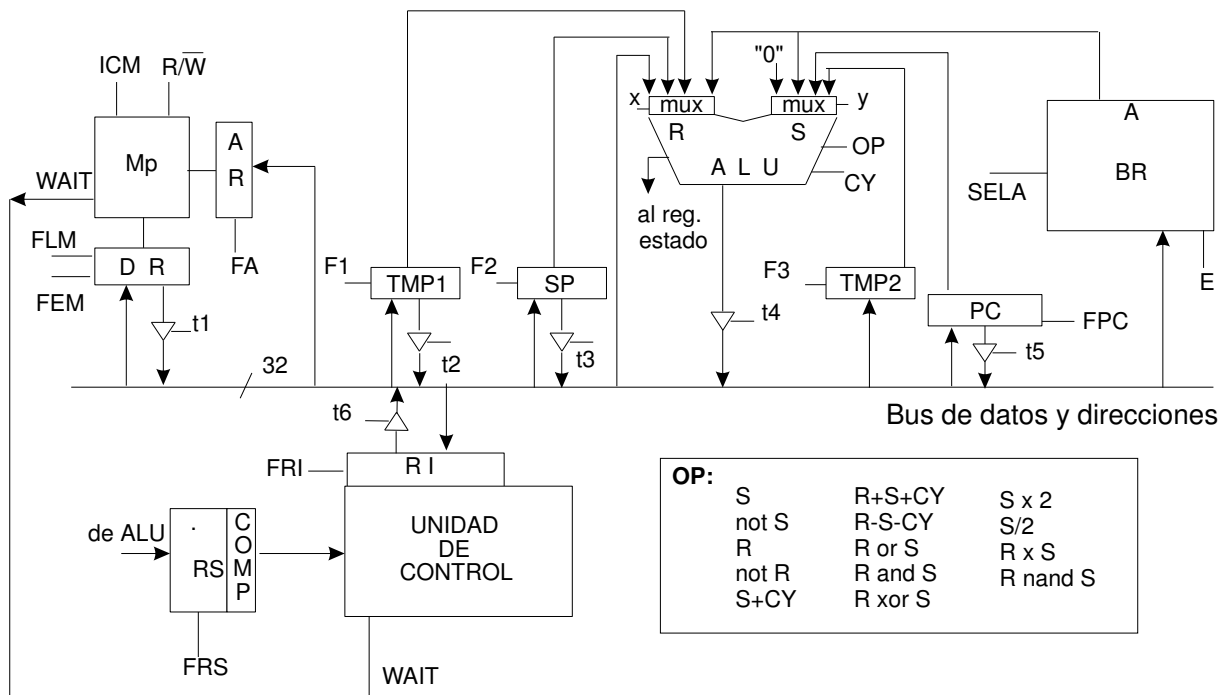


Figura 3 Estructura de la CPU y operaciones de la ALU

1. Considerando los siguientes tiempos:

- *Tiempo de lectura/escritura del banco de registros: 3 ns.*
- *Tiempo de la operación más lenta de la ALU: 10 ns.*
- *Tiempo de respuesta de una puerta triestado: 1 ns.*
- *Tiempo de lectura/escritura de los registros transparentes o específicos: 2 ns.*
- *Tiempo de acceso a memoria de control: 20 ns.*
- *Tiempo del secuenciador del microprograma: 3 ns.*

Determine el mínimo periodo de reloj que debe utilizarse en este computador, considerando despreciables los tiempos de lectura/escritura de los registros internos de la unidad de control.

2. *Microprograme a nivel RT (transferencia entre registros) la siguiente instrucción de una palabra, incluyendo la fase de fetch:*

CALL_ADDC .R4, [.R5 ++], \$ DESP

Esta instrucción, cuyo código de operación es de 12 bits, realiza la suma de los dos primeros operandos, almacenando el resultado en el primero y produce una llamada a subrutina si el resultado de la suma produce acarreo. La dirección de salto a subrutina, se especifica en el tercer operando con direccionamiento relativo a contador de programa, siendo DESP el correspondiente desplazamiento. Considere que existen en la máquina los mecanismos automáticos adecuados para hacer la extensión de signo del desplazamiento, cuando se carga en un registro.

3. *Determine el rango de direcciones en el que puede situarse la subrutina, teniendo en cuenta que la instrucción del apartado 2) está almacenada en la dirección de memoria $H'0FFFFFFF$ y que el desplazamiento está representado en complemento a 2. Justifique su respuesta.*

3 En la figura 4 se muestra esquemáticamente la estructura de una CPU de 32 bits cuya unidad de control es microprogramada con pipeline Registro de Control-Registro de Estado, en la que aparecen especificados los siguientes registros, ninguno de los cuales permite autoincremento o autodecremento:

- TMP1 y TMP2 son registros temporales transparentes al usuario.
- BR es un banco de registros compuesto por 18 registros generales (R1 a R18).
- AR y DR son, respectivamente, el registro de direcciones de memoria y el registro de datos de memoria.
- IR es el registro de instrucción, PC el contador de programa y SR el registro de estado.

En este computador, la pila crece hacia direcciones de memoria crecientes y el puntero de pila, SP, apunta a la primera dirección libre de la pila.

Los accesos a memoria son asíncronos, controlados por la señal WAIT.

Las operaciones que puede realizar la ALU están indicadas en el recuadro inferior de la figura 4.

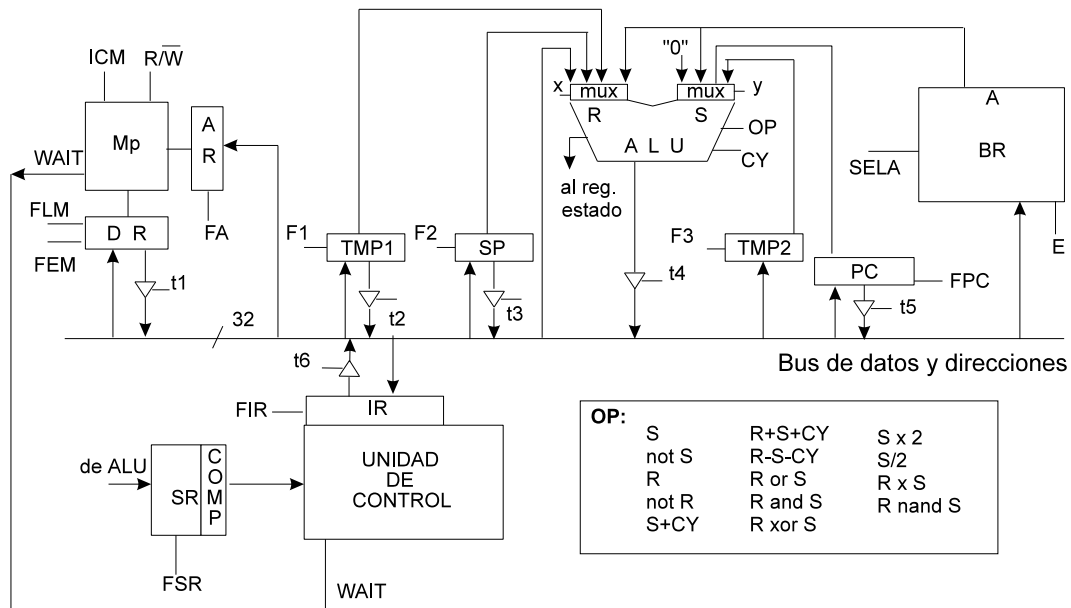


Figura 4

a) Teniendo en cuenta las especificaciones anteriores, realice los microprogramas a nivel RT (transferencia entre registros) de las instrucciones de una palabra PUSH [.Ri++] y POP [- -.Ri] y de la microsubrutina de fetch.

b) Calcule el tiempo total de ejecución de las instrucciones del apartado a), considerando que los accesos a memoria tienen una duración de 45 ns y que los retardos de los diferentes elementos de esta estructura son los siguientes:

- Lectura o escritura del banco de registros: 3 ns.
- Lectura o escritura de un registro: 2 ns.
- ALU y multiplexores: 14 ns.
- Buffer triestado: 1 ns.
- Secuenciador: 5 ns.
- Memoria de control: 10 ns.

4 Sea la CPU cuyo esquema simplificado aparece en la figura 5, en la que se muestran todas las rutas de datos y puertas triestado, aunque no las líneas de control/estado. El ancho de palabra es de 32 bits, así como todos los registros, rutas de datos y direcciones. El direccionamiento es a nivel de byte. La unidad de control es microprogramada.

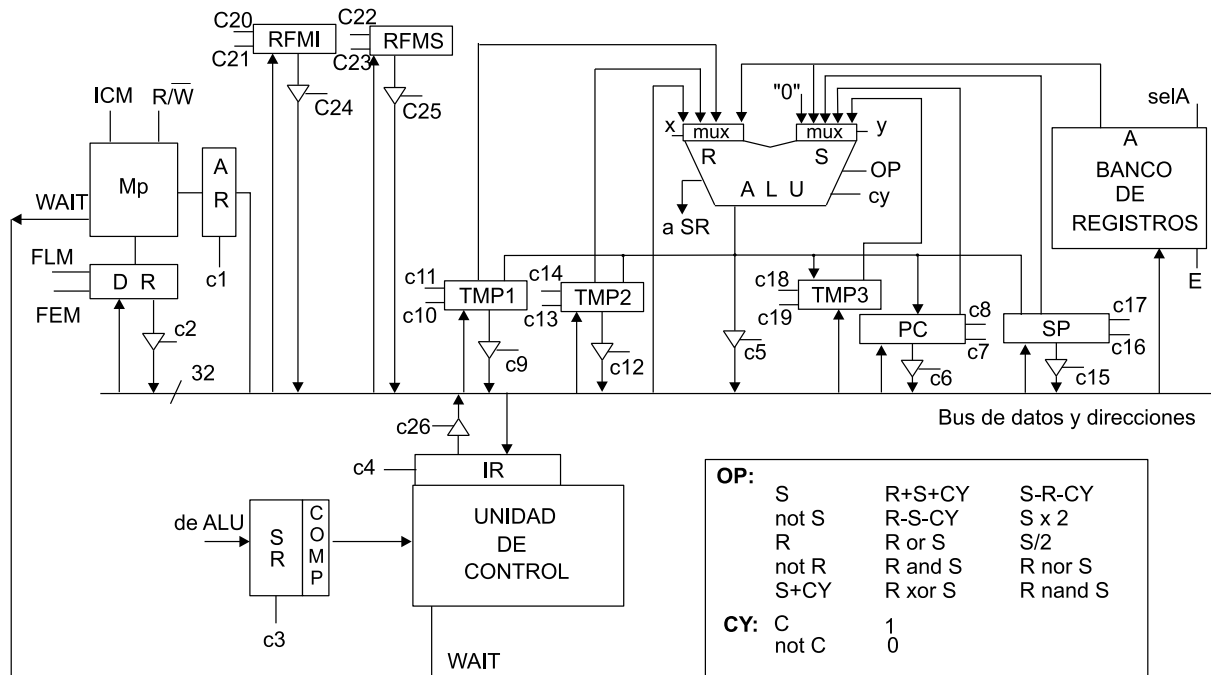


Figura 5

Los registros *TMP1* a *TMP3* son transparentes al usuario, y el banco de registros (*BR*) contiene dos registros transparentes más (*TMP4* y *TMP5*), y 14 registros de propósito general (*R0*, que siempre contiene 0, y *R1* a *R13*). La *ALU* puede hacer las operaciones indicadas en la tabla *OP* de la figura.

Esta CPU tiene dos modos de operación: modo supervisor y modo usuario. El registro *SR* es el registro de estado, y contiene, entre otros, el campo *S* (tendrá un 1 si opera en modo supervisor, y un 0 si opera en modo usuario).

Algunas operaciones de este computador están catalogadas como privilegiadas. En concreto, en modo usuario, la zona de memoria a la que se puede acceder tanto en lectura como en escritura está limitada por dos registros frontera, que apuntan a las direcciones mínima y máxima de memoria que el usuario puede direccionar (Registro Frontera de Memoria Inferior, *RFMI*, y Registro Frontera de Memoria Superior, *RFMS*). Cualquier intento de lectura o escritura en memoria fuera de este rango por parte de un usuario causará respectivamente una excepción de privilegio en lectura o una excepción de privilegio en escritura en memoria. En modo supervisor, se puede acceder a cualquier posición de memoria.

a) Considerando los siguientes tiempos:

- Tiempo de lectura/escritura de registros: 1 ut.
- Tiempo de retardo de triestados: 1 ut.
- Tiempo de retardo de decodificador: 3 ut.
- Tiempo de lectura/escritura de banco de registros: 3 ut.
- Tiempo de retardo de la *ALU*: 22 ut.
- Tiempo de acceso a Memoria de Control: 28 ut.
- Tiempo de retardo del secuenciador de microprograma: 5 ut.
- Tiempo de retardo de multiplexor: 2 ut.

Determine el mínimo tiempo de ciclo de reloj que debe utilizarse en este computador para los siguientes casos:

- 1) Que la unidad de control sea cableada.
- 2) Que la unidad de control sea microprogramada sin pipeline.

a.3) Que la unidad de control sea microprogramada con pipeline (con el tipo de pipeline más adecuado para este computador).

El juego de instrucciones del procesador incluye instrucciones de bloques de datos. En concreto, contiene la instrucción `COUNTGT #n, [.Ri], [.Rj], [.Rk]`.

La ejecución de esta instrucción lee el dato almacenado en $[\text{.Ri}]$, busca entre las n palabras apuntadas a partir de $[\text{.Rj}]$ los datos que sean mayores que el dato buscado ($[\text{.Ri}]$), y el número de datos encontrados que hayan resultado mayores, lo almacena en $[\text{.Rk}]$.

Suponga que existe una microrutina `comprobar_dir` que compara si la dirección que hay almacenada en AR es privilegiada o no, y deja en el biestable Z el resultado de si lo es ($SR.Z=1$), o no ($SR.Z=0$).

Escriba a nivel de transferencia entre registros, el microprograma de la instrucción `COUNTGT`, llamando a la microrutina `comprobar_dir`. Tenga en cuenta que esta instrucción puede causar violación de acceso en memoria. Suponga que también existen microsubrutinas que tratan las excepciones de privilegio en lectura de dato e instrucción en memoria, así como la de escritura en memoria.

Para este apartado, suponga que el computador tiene unidad de control microprogramada sin pipeline y que el tiempo medio de acceso a memoria es de dos ciclos.

b) Calcule el tiempo de ejecución de la rutina del apartado anterior suponiendo que la microrutina `comprobar_dir` tarda en ejecutar 12 ciclos de reloj, $n=1000$, no se producen excepciones, y:

b.1) La unidad de control es microprogramada sin pipeline.

b.2) La unidad de control es microprogramada con pipeline del tipo $RC-RD$ (Registro de Control-Registro de Direcciones).

5 La figura 6 muestra esquemáticamente la estructura de la CPU de un computador con direccionamiento a nivel de byte, cuyas palabras y direcciones son de 32 bits. La CPU tiene un bus interno de direcciones y otro bus interno de datos; un registro acumulador AC; registros temporales y transparentes al usuario T1, T2, T3 y T4; banco de registros BR con otros tres registros transparentes más (T10 a T12), además de 14 registros generales (R1 a R14), y un registro especial R0, que siempre contiene un 0. AR es el registro de direcciones de memoria, DR es el de datos de memoria, IR es el registro de instrucción, y SR es el registro de estado. La CPU dispone además del PC (contador de programa) y del SP (puntero de pila), en principio sin posibilidad de autoincremento ni autodecremento. La unidad de control es microprogramada, y dispone de un secuenciador con microcontador para microbucles, así como la posibilidad de realizar microsubrutinas.

Las operaciones que es capaz de realizar la ALU de este computador aparecen en el recuadro de la figura 6, y su memoria funciona de forma asíncrona, dependiendo de la activación de una señal Ready.

El juego de instrucciones de este computador tiene, entre otras, la instrucción de una palabra $ORM \#n, [.Ri], [.Rj]$, que realiza la operación OR del contenido de los dos vectores de n elementos apuntados por Ri y Rj , dejando el resultado a partir de Ri . Al finalizar la ejecución de esta instrucción, los registros Ri y Rj contendrán la dirección de comienzo de los vectores.

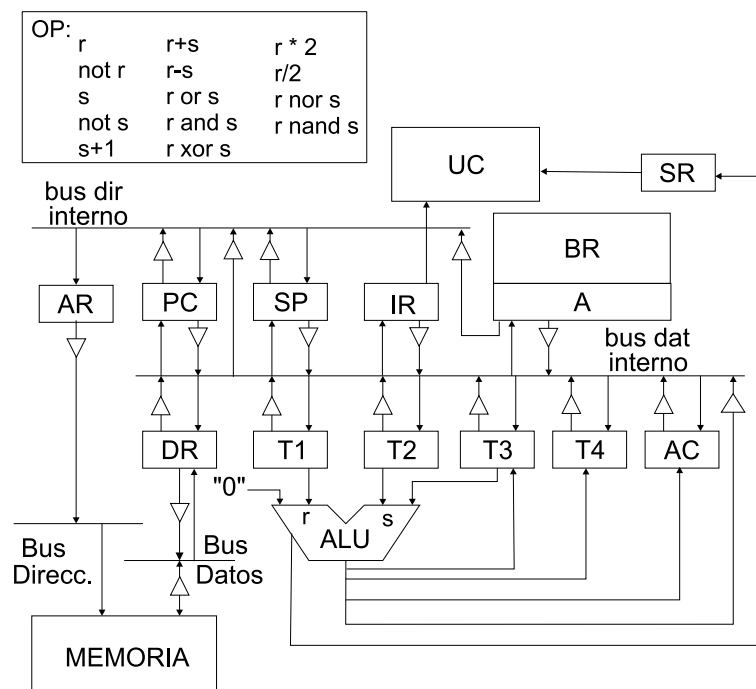


Figura 6. Estructura de la CPU y operaciones de la ALU

a) Teniendo en cuenta que el direccionamiento es a nivel de byte, indique posible/s mejora/s en el diseño de la estructura que mejore el rendimiento. Explique y argumente su respuesta.

b) Modifique la estructura de la CPU de la figura para que el rendimiento sea mejor en la instrucción descrita en el enunciado, describiendo las mejoras de sus propuestas.

c) Micro programe a nivel RT (transferencia entre registros) la instrucción del enunciado (incluyendo la microrrutina de fetch) haciendo uso de las modificaciones introducidas en el apartado a).

d) Suponga que en el juego de instrucciones existe también la siguiente instrucción de una palabra $ORCMPZM \#n, [.Ri], [.Rj], [.Rk]$

Esta instrucción es similar a la anterior ORM , ya que también realiza la operación OR del contenido de los dos vectores de n elementos apuntados por Ri y Rj , dejando el resultado a partir de Ri . Pero al mismo tiempo, va elaborando un nuevo vector en memoria a partir de la dirección apuntada por Rk . En este otro vector incluye un 0 en la posición de memoria correspondiente cuando la operación OR da como resultado un 0, y en caso contrario, almacena un 1.

Indique a grandes rasgos cómo se podría perfilar la microprogramación de la instrucción $ORCMPZM$, partiendo de un código similar al del apartado c), modificándolo. Por ejemplo, añadiendo llamadas a microsubrutinas. Indique dónde, y en qué consistirían éstas, y si habría que cuidar información guardada en registros temporales de alguna forma, cómo se podría salvarguardar, etc.

6 Sea la CPU cuyo esquema simplificado aparece en la figura 7, en la que se muestran las rutas de datos, puertas triestado, y las señales de control/estado. El ancho de palabra es de 32 bits, así como todos los registros, rutas de datos y direcciones. El direccionamiento es a nivel de byte.

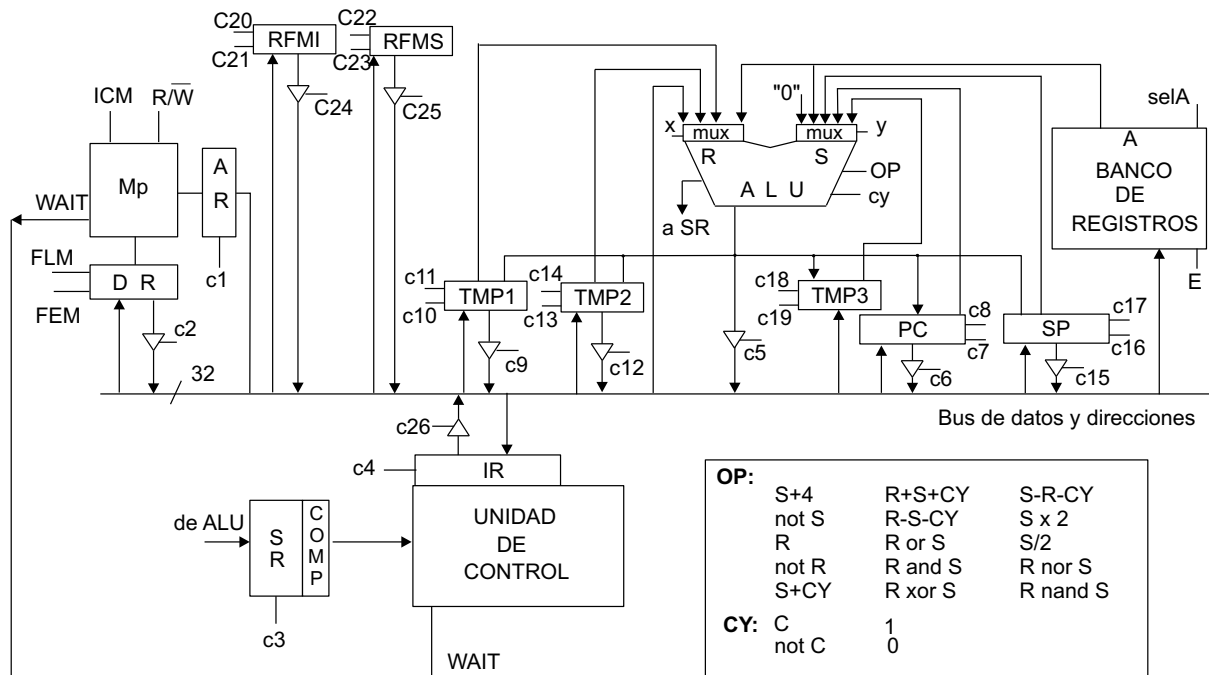


Figura 7

Los registros *TMP1* a *TMP3* son transparentes al usuario, y el banco de registros contiene dos registros transparentes más (*TMP4* y *TMP5*), y 14 registros de propósito general (*R0*, que siempre contiene 0, y *R1* a *R13*). La *ALU* puede hacer las operaciones indicadas en la tabla *OP* de la figura.

Esta CPU tiene dos modos de operación: modo supervisor y modo usuario. El registro *SR* es el registro de estado, y contiene, entre otros, el campo *S* (tendrá un 1 si opera en modo supervisor, y un 0 si opera en modo usuario).

Algunas operaciones de este computador están catalogadas como privilegiadas. En concreto, en modo usuario, la zona de memoria a la que se puede acceder tanto en lectura como en escritura está limitada por dos registros frontera, que apuntan a las direcciones mínima y máxima de memoria que el usuario puede direccionar (Registro Frontera de Memoria Inferior, *RFMI*, y Registro Frontera de Memoria Superior, *RFMS*). Cualquier intento de lectura o escritura en memoria fuera de este rango por parte de un usuario causará respectivamente una excepción de privilegio en lectura o una excepción de privilegio en escritura en memoria. En modo supervisor, se puede acceder a cualquier posición de memoria.

Considerando los siguientes tiempos:

- Tiempo de lectura/escritura de registros: 1 ut.
- Tiempo de retardo de triestados: 0,5 ut.
- Tiempo de retardo de decodificador: 2 ut.
- Tiempo de lectura/escritura de banco de registros: 2 ut.
- Tiempo de retardo de la ALU: 25 ut.
- Tiempo de acceso a Memoria de Control: 15 ut.
- Tiempo de retardo del secuenciador de microprograma: 8 ut.
- Tiempo de retardo de multiplexor: 2 ut.
- Tiempo medio de acceso a memoria principal: 55 ut.

a) Determine el mínimo tiempo de ciclo de reloj que debe utilizarse en este computador para los siguientes casos:

1. Que la unidad de control sea cableada.
2. Que la unidad de control sea microprogramada sin pipeline

3. Que la unidad de control sea microprogramada con pipeline (con el tipo de pipeline más adecuado para este computador).

b) Microprograme una microsubrutina `comprobar_dir` que compruebe si la dirección en la que se va a operar en memoria es privilegiada o no. Indique y decida cuáles son y dónde se dejan tanto el parámetro de entrada como el de salida en esta subrutina.

c) Se quiere incorporar al repertorio de instrucciones de esta CPU la instrucción **privilegiada de dos palabras** `LD .SR, /Dir`. Se pide:

1. Modifique la estructura de esta CPU para que se pueda ejecutar dicha instrucción.
2. Escriba a nivel de transferencia entre registros, el microprograma de la instrucción `LD .SR, /Dir` teniendo en cuenta que el computador tiene unidad de control microprogramada con pipeline del tipo RC-RD (Registro de Control-Registro de Direcciones). Utilice la microsubrutina `comprobar_dir` si cree que le hiciera falta. Tenga en cuenta que esta instrucción puede causar violación de privilegio por código de operación por tratarse de una instrucción privilegiada (ya que pudiera escribir en el registro de estado SR). Incluya la microrutina de fetch.
3. Teniendo en cuenta sus respuestas al resto de este apartado, calcule el tiempo de ejecución de la instrucción anterior suponiendo que se ejecuta con $SR.S=1$.

7 Sea la CPU cuyo esquema simplificado aparece en la figura 8, en la que se muestran las rutas de datos, puertas triestado, y las señales de control/estado. El ancho de palabra es de 32 bits, así como todos los registros, rutas de datos y direcciones. El direccionamiento es a nivel de byte.

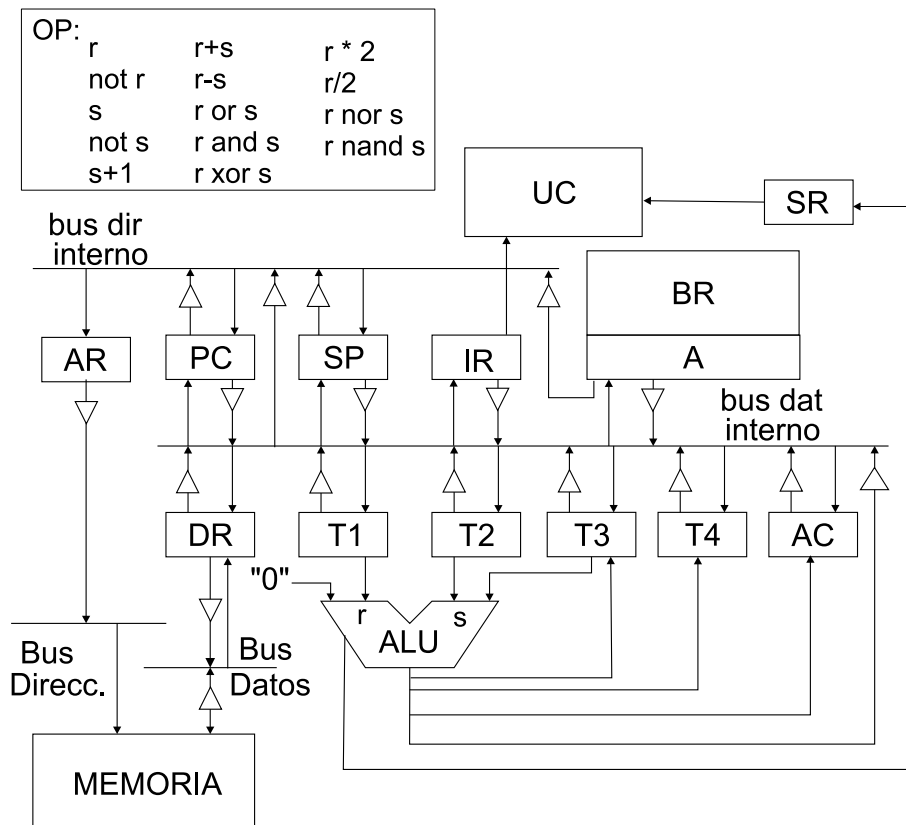


Figura 8

Los registros T1 a T4 son transparentes al usuario, el AC es un registro Acumulador, PC es el Contador de Programa, SP es el Puntero de Pila y el banco de registros contiene dos registros transparentes más (T5 y T6), y 26 registros de propósito general (R0, que siempre contiene 0, y R1 a R25). La ALU puede hacer las operaciones indicadas en la tabla OP de la figura.

La Unidad de Control es microprogramada, y el tiempo de ciclo de reloj es de 100 ut. El tiempo medio de acceso a memoria principal: 150 ut.

a) En el repertorio de instrucciones se quiere incluir la llamada a subrutina condicional:

CALLEQ .Ac, .Ri, [.Rj]

Esta instrucción salta a la subrutina almacenada a partir de [.Rj] en el caso de que el contenido del Acumulador y el registro .Ri sean iguales. Si no, continúa secuencialmente.

1. Microprogramme a nivel RT (Transferencia entre Registros) la instrucción CALLEQ. Incluya el microcódigo del ciclo de fetch.
2. ¿Cuánto tarda en ejecutarse esta instrucción (incluido el fetch), suponiendo que la probabilidad de salto es del 80 %? (tenga en cuenta su respuesta del apartado anterior)

b) Se quiere modificar el diseño previo de esta CPU para que tenga dos modos de operación: modo supervisor y modo usuario, de forma que, se pueda controlar que haya instrucciones y ciertas operaciones no permitidas en modo usuario, por ser privilegiadas. En concreto, acceder a memoria fuera del rango permitido sería una de estas operaciones privilegiadas que sólo podría hacerse en modo supervisor. Cualquier intento de lectura o escritura en memoria fuera de este rango por parte de un usuario causará respectivamente una excepción de privilegio en lectura o en escritura en memoria.

1. *Modifique la estructura de esta CPU para que se puedan controlar las excepciones de acceso a zona privilegiada de memoria. Indique cuál es el mecanismo para recordar cuál es el modo de ejecución actual, y cómo se podría modificar este modo. ¿Sería esta modificación una operación privilegiada, o una operación válida al usuario?*
2. *Indique qué modificaciones habría que hacer en el microprograma que propuso de la instrucción `CALLEQ .Ac, .Ri, [.Rj]` para que compruebe posibles excepciones de acceso a zona privilegiada de memoria. ¿Cómo afectarían estas modificaciones al tiempo de ejecución de esta instrucción?*