

Ej (2)

- (a) Si una falla de hardware hiciera que la salida del FF menos significativo del MIP quedara conectada permanentemente a cero el campo que se veía afectado sería el Jump Address.

Este campo indica cual será la próxima microinstrucción a ejecutarse. Contiene 11 bits, y por lo tanto, si el último bit siempre es 0, las direcciones no podrán ser impares, ya que pare que lo sean requieren que el bit menos significativo sea 1. Entonces no se podría acceder a direcciones pares salvo que sea utilizando la cond correspondiente a NEXT, ya que ignora la dir del JUMP ADDR.

- (b) Una estructura de bus tiene propósito de reducir la cant de interconexiones entre todos los componentes disponibles a un conjunto de cables específico. Estos cables se subdividen en: bus de datos, bus de control y bus de direcciones. De esta forma se logra comunicar todos los componentes de una computadora.

El bus de datos contiene la información que se quiere transmitir de un dispositivo a otro.

El bus de dir contiene la dir a donde se quieren llevar los datos.

El bus de control es el que controla, maneja y ordena los otros dos buses.

Ejercicio ③

① Puede el programa optar por llamar subrutinas que estén en el mismo módulo o que estén en otro módulo si decide programar la subrutina en el mismo módulo lo debe hacer dentro de las directivas .begin y .end. No es necesario que defina la rutina antes de llamarla, ya que el ensamblador de ARC es de dos pasadas.

Si en cambio deseara llamar a una subrutina definida en otro módulo que el programa principal, el programa principal a priori no va a reconocer dicha rutina, ya que no se definió en el mismo módulo es por eso que se deben utilizar las pseudo-operaciones para que la subrutina fue definida en otro módulo, que el linker más tarde unirá a este programa.

Las pseudo-operaciones / Directivas son:

- extern : es la etiqueta que se escribe adelante del nombre de la sub en el programa principal para indicarle ^{al ensamblador} que está en otro módulo.

- global : se antepone el nombre de la sub en el módulo donde se encuentra desarrollada, para indicarle al ensamblador, que otros módulos pueden llamarla.

sin estas directivas, el programa no funcionara ya que se dara un error de ensamblado cuando no encuentre a la sub en el modulo.

(b) conveniencia de constantes → lo respondi en un final de los anteriores.

Ejercicio (4)

(a) minimizar numero de fallos de caché.

Un fallo de caché se considera a cuando se quiere acceder a una variable que no ha sido copiada a caché, por lo que hay que acceder a RAM. Para evitar el acceso a RAM, tenemos que aprovechar el principio de localidad.

Por lo tanto, para minimizar la cantidad de fallos de caché se me ocurre:

- utilizar código recursivo e iterativo
- utilizar las mismas variables lo maximo que se pueda
- no realizar saltos de memoria.

⑥ manejador de memoria o MMU

La MMU surge como una alternativa del loader que se basa en las técnicas de administración de memoria, realiza la ubicación mediante la carga, en un registro base de segmento, de la base apropiada para la ubicación del código en el correspondiente lugar de memoria física. La Unidad de administración de memoria (MMU) suma el contenido de este registro base a todas las referencias de memoria. Como resultado, cada programa puede iniciar su ejecución en la dirección de memoria cero y confiar en la MMU para lograr la reubicación de todas las referencias de memoria en forma transparente.