

Estructura del computador

66.70

Primer cuatrimestre 2014

Federico L. Amura

95202

Enunciado

Se propone diseñar la lógica de un reproductor de DVD para controlar la apertura y cierre de la unidad y la reproducción del vídeo. El operador cuenta con cuatro botones: Eject, Play, Stop y Pausa. El sistema, en función de estos botones, deberá encender y apagar el motor, abrir o cerrar la bandeja así como controlar los modos reproducción, detenido y pausa.

El problema debe ser formalizado definiendo sus entradas, salidas y un diagrama de estados que describa su funcionamiento. Definir claramente todos los estados y las condiciones de transición entre ellos.

Un sensor electrónico detecta la presencia de un disco en el equipo manteniendo un 1 lógico a su salida en tal caso. El mecanismo de apertura y cierre de la puerta asegura con un 0 lógico a su entrada que la puerta estará cerrada y con un 1 lógico que la misma estará abierta.

Considerar que cuando el disco está en movimiento la apertura de la bandeja sólo puede realizarse tras esperar 3 segundos para que el motor detenga su giro.

Se entiende que en el modo pausa el disco sigue girando pero se suspende la lectura de datos.

Se piden dos soluciones:

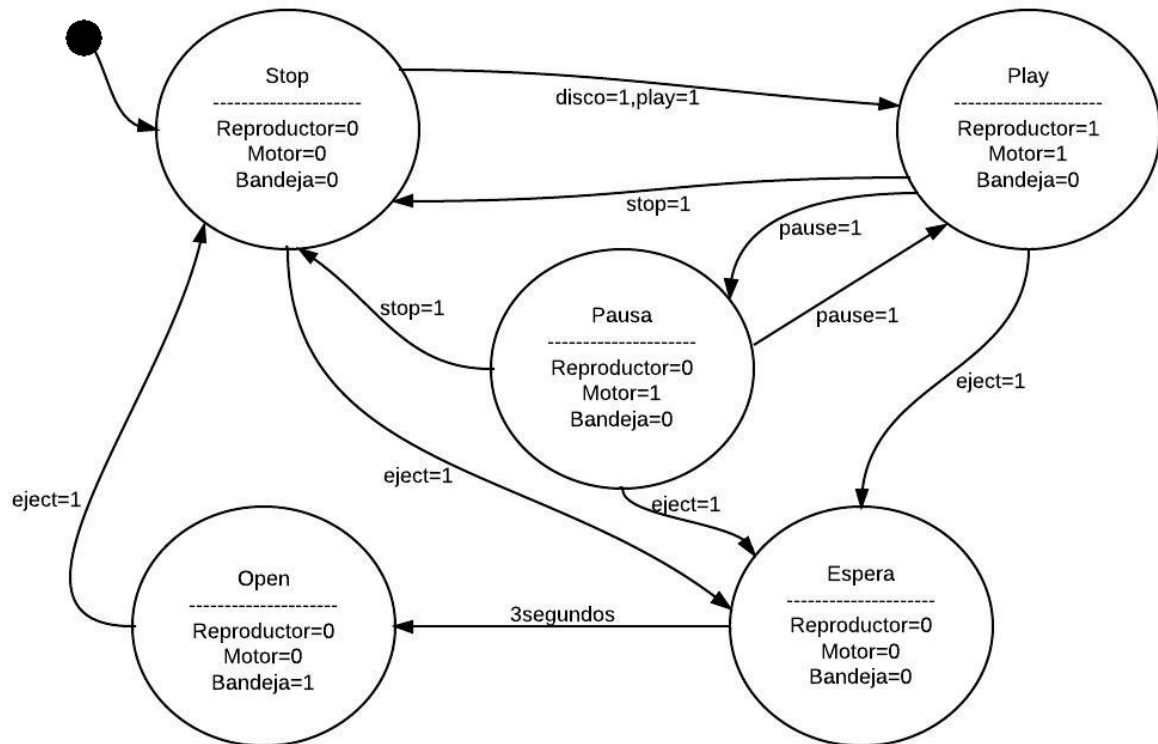
- Solución cableada: circuito lógico dedicado específicamente a esta función y que está formado por compuertas, FF, etc.
- Solución basada en un microprocesador: software en Assembly de un sistema ARC

Entradas y salidas del sistema

Las entradas del sistema son 4. Los 3 botones, play, pausa y stop. Además está el sensor de disco. Entre estas 4 se definen las transiciones que realiza el sistema considerando el estado en el que está.

Las salidas son las 3 cosas que se controlan, el reproductor, el motor que hace girar el disco y la bandeja del disco.

Diagrama de estados



Consideraciones

- El sensor de disco está en 1 cuando sensa que tiene un disco dentro
- Para evitar la superposición de instrucciones, los botones solo ingresa el pulso, es decir, si aprieto play, recibo un pulso que indica la orden sin importar si sigo o no apretando el botón
- El reproductor hace las lecturas de los pulsos en los flancos ascendientes. Para evitar la pérdida de un comando en un flanco el pulso debe ser más largo que el ciclo del clock
- Se elige un clock lo suficientemente rápido como para que el usuario, al querer apretar dos botones al mismo tiempo no pueda hacerlo dado la inexactitud de oprimir dos botones juntos con la mano
- Si el disco está girando y se lo desea extraer entonces la bandeja debe esperar a que el motor se detenga para abrir

- En pausa se detiene la reproducción, no así el motor.
- En stop se detienen la reproducción y el motor

Transiciones

Dado que el circuito es muy simple y tiene ciertos estados definidos con sus salidas en un estado concreto, sin importar el estado previo o el siguiente, no se realiza un estudio de transiciones profundo. Mas bien el funcionamiento se basa en que llegada una orden, si esta es aceptada, se pasa al estado correspondiente, poniendo las salidas como corresponden.

De esta manera, por ejemplo, si el circuito tiene un disco dentro, esta detenido, es decir, sin reproducir y sin girar el motor, si además no se encuentra contando para quitar el disco entonces el sistema puede aceptar la orden play, que pone el motor a girar y el reproductor a funcionar. De la misma manera ocurre con las otras ordenes, solo cambia las consideraciones que tienen para ser aceptadas y los estados en los que ponen las salidas.

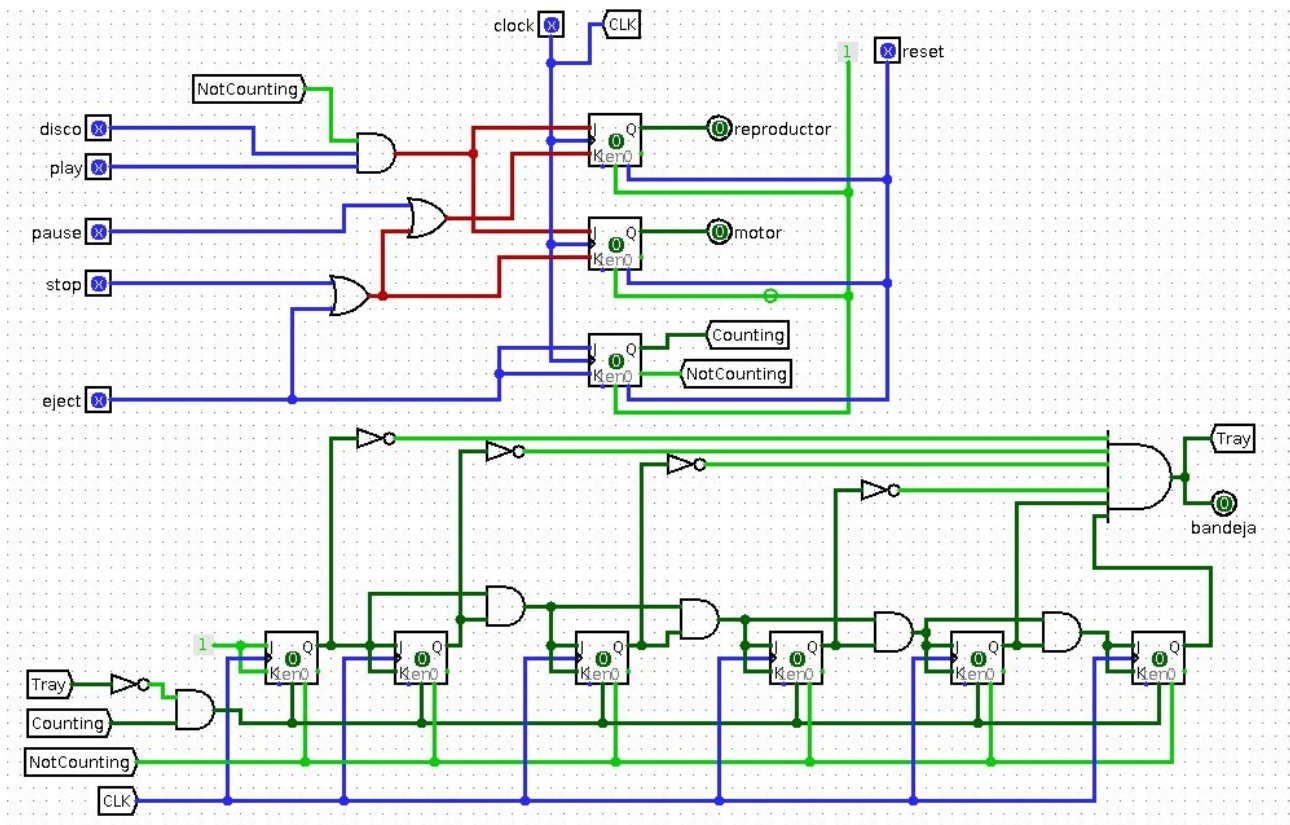
La única excepción a esta metodología vendría a ser el meta estado Espera, el cual tiene las mismas salidas que Stop. Este estado cumple una sola función que es asegurarse que el motor pare de girar antes de abrirse la bandeja del disco. Por lo cual vendría a ser un precursor del Open o mas bien un modo de pasar a ese estado. Difiere de los demás en que el mismo se cambia hacia otro estado automáticamente luego de cumplir cierto tiempo sin esperar ninguna orden de parte del usuario..

Codificación de estados

Como se puede ver en el diagrama de estados, la codificación esta definida según lo que se requiere que funcione. Los estados serian las columnas y las salidas las filas.

	Play	Pausa	Stop	Open	Espera
Bandeja	0	0	0	1	0
Motor	1	1	0	0	0
Reproductor	1	0	0	0	0

Solución cableada



Como se ve en la imagen se puede dividir el circuito en dos partes.

La parte inferior es un simple contador binario sincrónico, que se alimenta de los pulsos de un clock definido tomando en cuenta los supuestos, en este caso de 32Hz. Está controlado por el tercer FF-JK del circuito, el cual cuando tiene su salida en 0 (y por lo tanto la negada en 1) mantiene todos los FF del contador en cero y no lo habilita a contar. Luego de cambiar su estado, ya no sigue imponiendo un 0, pasa a habilitar los contadores y estos cuentan. Llegado el valor 96 (ya que $96/32\text{hz}=3\text{seg}$ de retardo para que el motor pueda detenerse) se abre la bandeja, y gracias a la NAND este contador deja de aumentar, quedando en ese valor. Luego de cambiar de nuevo el estado del FF-JK que controla este contador se reinicia el contador, se desactiva la salida de la bandeja y vuelve todo al estado inicial (contador y circuito completo).

La otra parte del circuito sería la controladora propiamente dicha, esta se basa en 3 FF-JK para cada una de las salidas del sistema. El primero, encargado de manejar la reproducción, solo se activa, gracias a una compuerta AND cuando el sensor de disco está activo, el contador se encuentra desactivado y se le da al botón play. La pausa, simplemente detiene la reproducción, no es necesario que realice ninguna evaluación. Hay que destacar que stop y eject cuando son activados también ejecutan la pausa. El segundo flipflop, maneja el motor, este solo se activa con la reproducción, pero a diferencia de este no se detiene con la pausa si no que con el stop y el eject. Todo esto está controlado con un par de compuertas OR. El tercero de los flipflops, es en encargado de arrancar el contador.

Lista de materiales de la solución cableada.

Para esta solución se necesitarían:

- 9 flip flops JK (\$6)
- 1 AND de 6 entradas (\$6)
- 1 AND de 3 entradas (\$3)
- 5 AND de 2 entradas (\$2)
- 2 OR de 2 entradas (\$2)
- 4 NOT (\$1)
- 4 Entradas (\$10)
- 3 Salidas (\$10)

Sumando un subtotal de \$151, mas lo trabajado, unas 3 horas-hombre en el analisis, diseno e implementación a \$200 cada una nos da un total final de \$751.

Hay que destacar que esta lista de materiales es solo para esta solución, se pueden realizar infinidad de cambios ya sea para reducir componentes, cambiarlos, etc. Además, el contador así como esta puede reemplazarse con un contador ya incorporado en un único componente.

Solución programada

El código va anexo en formato .asm

La solución programada es en cierta manera parecida pero difiere en sus métodos ya que ahora el sistema se encuentra en un estado definido y realiza las comprobaciones que competen a ese estado.

La forma mas fácil de ver el funcionamiento es realizar un seguimiento de un caso particular. Iniciando el programa, este se carga en el estado Stop (etiqueta reset del programa), el sistema pone todas las salidas en cero (con los comandos ld, andn y st), y luego entra en un ciclo (etiqueta parado) en el que se lee el registro de las entradas (ld hacia r1), se toman los valores que pueden hacer que se cambie el estado (and con EJECT+PLAY) y luego se verifica si la entrada EJECT estaba activada (andcc), si el flag zero no se hubiese activado, indicaría que fue oprimido, por lo que saltaría al cambio de estado correspondiente (mediante el bne esperando). Particularmente este estado, realiza una comprobación ahora de la misma manera con el flag del sensor de disco (ldd y andcc) el cual de dar cero (no tener un disco) no permite que se realice otra comprobación y vuelva a iniciar el ciclo (be parado), solo pudiendo aceptar la orden Eject. De tener un disco, este branch no se ejecuta y de la misma forma que se fijo si se ingreso el comando Eject ahora lo hace con el comando Play (andcc y bne). De no haberse oprimido tampoco play, vuelve a repetir el ciclo completo.

El funcionamiento es similar para todos los estados, se entra en ellos, se setean las salidas correspondientes y luego se entra en un ciclo para determinar el siguiente salto. La excepción otra

vez seria el meta estado de espera para abrir la bandeja, donde primero apago el motor y el reproductor (con ld, andn y st), luego se carga la cantidad de veces que se tiene que repetir el ciclo (ld TRESSEGUNDOS), se repite el decremento hasta cumplir el tiempo (addcc y bne) y finalmente se pasa por abrircerrar, abriendo o cerrando la bandeja (xor para cambiar solo el valor de la bandeja, st para aplicar el cambio) y se vuelve al estado de reset ya sea con la bandeja abierta o cerrada (mediante el ba).

Lista de materiales de la solución programada.

Para esta solución se necesitarían:

- 1 microprocesador ARC (\$1000)

Sumando un subtotal de \$1000, mas lo trabajado, a un precio de \$10 por linea de código, y 70 lineas aproximadamente, daría un total de \$1700.

Al igual que con la solucion cableada, hay que aclarar que esta es una de las muchas soluciones posibles en las que pueden aplicarse infinidad de cambios.

Comparación

Una primera comparación surge viendo los precios de ambas soluciones, siendo muy superior el de la programada, la cual supera a la cableada con solamente sus componentes y casi tambien con el valor del trabajo.

Esta comparación es un fiel reflejo de la complejidad de ambas soluciones, la cableada es muy facil de comprender en una mirada rápida, o con una breve explicación como mucho, en cambio la programada requiere una explicación mas avanzada y una mirada mas intensa dada su mayor complejidad.

No solamente se refleja de esa manera si no que también se vio la diferencia de complejidad en las horas trabajadas, haciendo que la solución programada se logre luego de mas tiempo.

Otros detalle a considerar podría ser el agregado de alguna nueva función o cambio en las que se tienen, si se llegara a tener que aplicar cambios, dada su menor complejidad es mejor aplicarlos sobre la solución cableada en vez de la programada.

Con estas consideraciones es fácil tener preferencia para la solución cableada en vez de la programada.