

Exercicis Tema 9

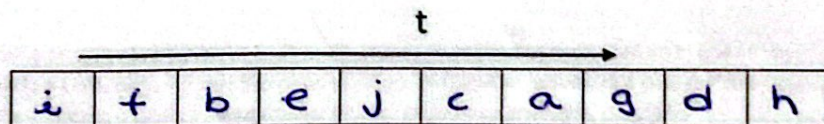
S'ha d'entregar un únic fitxer PDF que inclogui la solució que vosaltres proposeu als problemes plantejats. El fitxer PDF no te que ser necessàriament una solució feta per ordinador, pot ser una solució escrita a mà i digitalitzada. El PDF ha d'incloure una capçalera on s'indiqui el vostre nom i cognoms, i l'enunciat de cada pregunta abans de la vostra resposta.

S'han de escollir i fer només 4 dels 6 exercicis proposats (a elecció de cadascú).

Exercici 1

Ordena temporalment les següents accions perquè es compleixi el protocol de handshaking en una comunicació asíncrona bidireccional entre un computador i un coprocessador. El processador té una entrada de control externa anomenada Begin que l'indica (quan es posa a 1) que s'ha de començar una nova operació. Per cada nova operació el processador envia dada a través del bus X al coprocessador. Com que el processador i el coprocessador estan controlats per senyals de rellotge diferents, aquests es comuniquen a través d'un protocol de comunicació asíncrona de handshaking de 4 fases. El coprocessador recull la dada, la processa calculant el resultat d'una funció matemàtica F sobre aquest valor i retorna el resultat al processador a través del bus Y:

- El coprocessador calcula el resultat d'aplicar la funció F sobre la dada recollida.
- El computador posa el senyal Req a 1.
- El computador posa el senyal Req a 0, deixant de mantenir la dada al bus X.
- El computador detecta el resultat correcte al bus Y i el captura.
- El coprocessador detecta que hi ha una dada correcte en el bus X i la captura.
- El computador posa la dada al bus X.
- El coprocessador posa el resultat de l'operació al bus Y.
- El coprocessador detecta que el computador ha llegit correctament el resultat, posant el senyal Ack a 0.
- El computador (emissor) espera que el senyal Ack valgui 0 i el senyal Begin 1.
- El coprocessador posa el senyal Ack a 1.



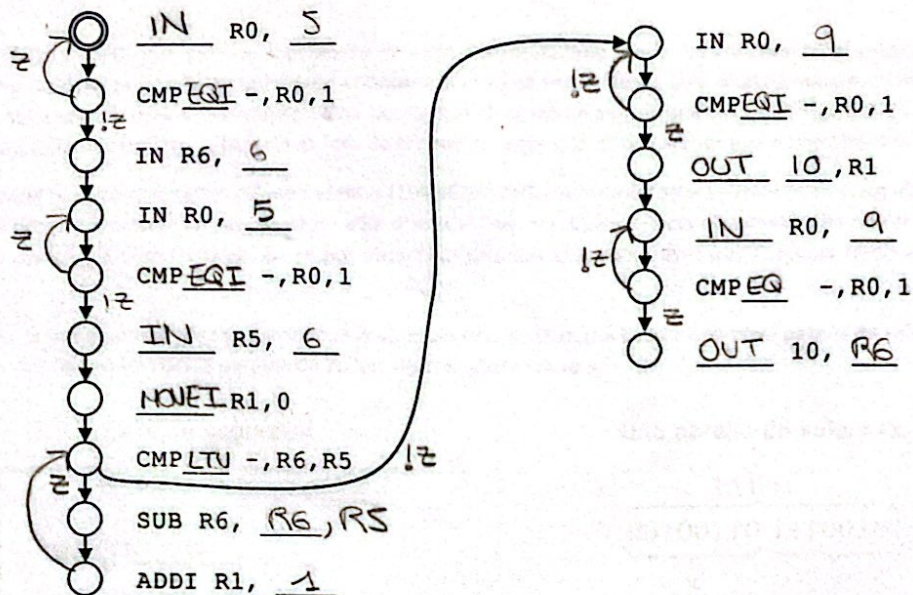
Exercici 2

Se desea resolver el problema de la división usando un PPE pero ahora usando la UPG. Se desea calcular el cociente y el resto de la división de dos naturales mayores que 0. Dados n (numerador) y d (denominador) tal que $n > 0$ y $d > 0$, calcula los naturales q y r que cumplen que $n = d * q + r$, y $0 \leq r < d$ mediante restas sucesivas.

Se ha conectado a la UPG un dispositivo externo de entrada que nos envía valores naturales de 16 bits y que tiene el registro de status en la dirección 5 del espacio de direccionamiento de entrada y el de datos en la 6. Este dispositivo tiene un efecto lateral en la lectura del dato sobre su registro de estado.

Este dispositivo de entrada nos envía de forma asíncrona primero n (el numerador) y luego d (el denominador). Una vez recibidos los dos valores deberemos calcular q (el cociente) y r (el resto) de la división haciendo restas sucesivas.

Usando la UPG junto a una UCe, una vez realizada la división se deberán enviar los resultados q y r (primero q y seguidamente r) a un dispositivo externo de salida con efecto lateral en el puerto de datos. Este dispositivo tiene el registro de status en la dirección 9 de entrada y el de datos en la 10 del espacio de direccionamiento de salida. Completad el grafo de estados si estuviésemos utilizando una unidad de control específica (UCe) junto a la UPG para que realice la función anteriormente descrita. Indicad los arcos y las etiquetas de los arcos (z , $1z$, o nada) que falten (en caso que falten) y completad las casillas de cada palabra de control.



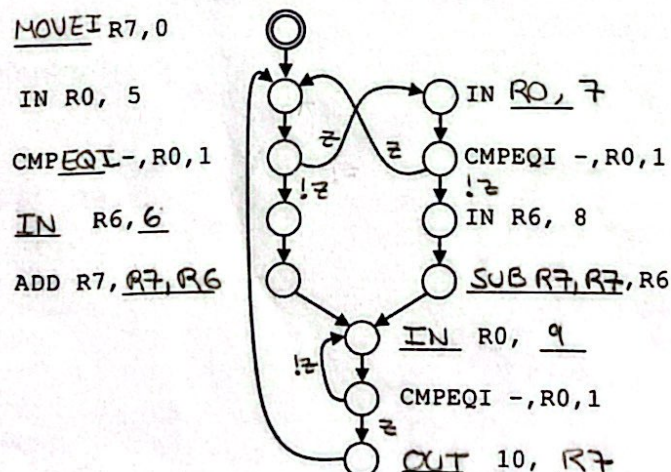
Exercici 3

Se han conectado a la UPG dos dispositivos externos de entrada (*dispA* y *dispB*) que nos envían valores enteros y un dispositivo externo de salida (*dispIMP*) al cual le podemos enviar datos. Todos estos dispositivos tienen un efecto lateral sobre su registro de estado.

Se desea realizar una función que vaya leyendo los datos de ambos dispositivos en el orden en el que se vayan detectando. Partiremos de un registro inicializado a cero, cada vez que llegue un valor por el *dispA* se le sumará a este registro y cada vez que llegue un valor por el *dispB* se le restará. A cada actualización del valor de este registro se enviará al *dispIMP*.

Suponiendo que el dispositivo *dispA* tiene el registro de status en la dirección 5 del espacio de direccionamiento de entrada y el de datos en la 6; que el *dispB* tiene el registro de status en la dirección 7 y el de datos en la 8; y que el *dispIMP* tiene el registro de status en la dirección 9 de entrada y el de datos en la 10 del espacio de direccionamiento de salida, completad:

El grafo de estados si estuviésemos utilizando una unidad de control específica (UCE) junto a la UPG para que realice la función anteriormente descrita. Indicad los arcos y las etiquetas de los arcos (z, !z, o nada) que falten (en caso que falten) y completad las casillas de cada palabra de control.

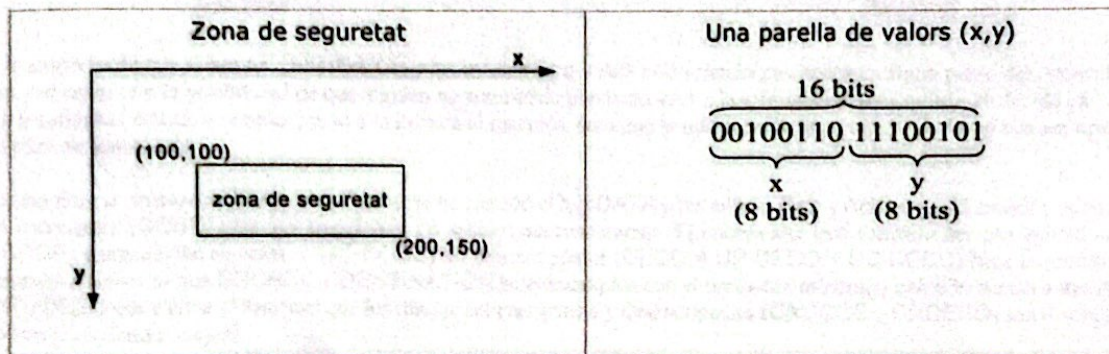


Exercici 4

Dissenyem un PPE (UCE+UPG) que amb un subsistema de entrada/sortida amb efecte lateral controli el sistema de seguretat d'un museu. El PPE ha de detectar si una determinada coordenada (x,y) es troba dins o fora d'una àrea determinada. La coordenada correspon a un objecte molt valuós del museu, i s'ha obtingut mitjançant un sensor que detecta la posició d'un dispositiu emissor incorporat a la peça del museu. Quan l'objecte és fora de la zona de seguretat, s'ha d'activar una alarma (fent que valgui 1).

La zona de seguretat la delimita el rectangle amb vèrtexs (100,100) i (200,150) i amb costats paral·lels als eixos de coordenades. Quan la peça es troba just al perímetre del rectangle no s'ha d'activar l'alarma. Quan s'inicia el sistema s'ha de fer que l'alarma estigui desactivada. Un cop s'ha activat l'alarma, no es pot desactivar fins que s'ha comprovat que l'objecte torna a ser dins la zona de seguretat.

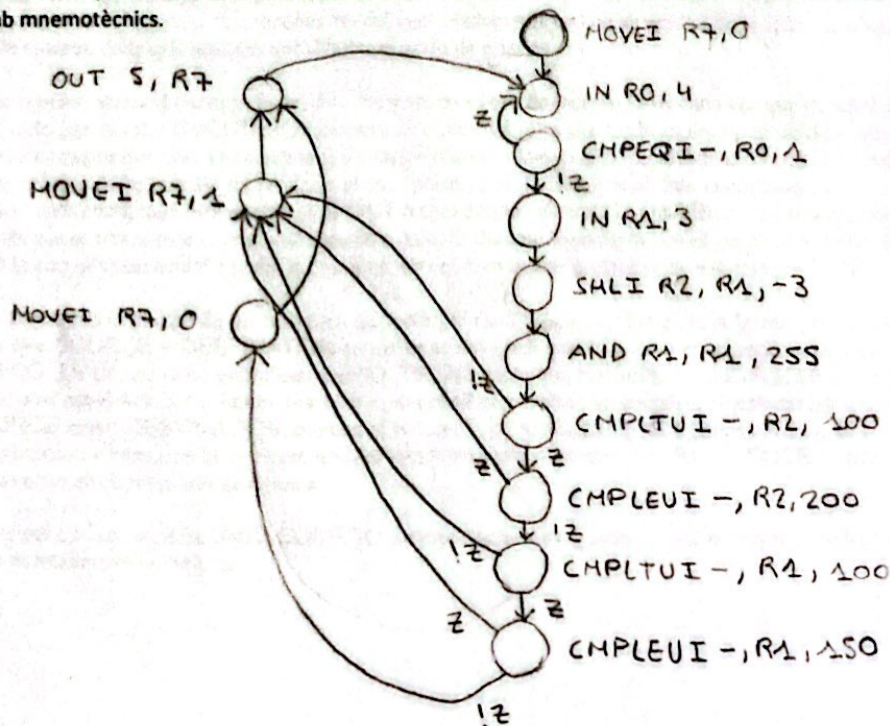
Quan acabi de fer la comprovació d'una coordenada (x,y), el sistema ha d'anar a buscar una nova parella de valors (x,y). Una parella de valors (x,y) es codifica en una única paraula de 16 bits de la següent manera:



Els 8 bits de més pes de la paraula es corresponen a la coordenada x i els 8 bits de menys pes a la coordenada y. Aquests dos valors estan codificats en binari natural amb 8 bits. Si en algun instant el sensor no és capaç de detectar l'objecte, genera la coordenada (0,0).

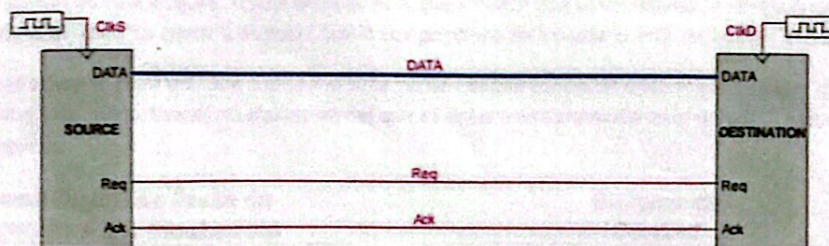
El dispositiu amb els sensors que ens proporciona la parella de valors (x,y) es connecta al port 3 d'entrada (INPUT[3]) mentre que al port 4 d'entrada (INPUT[4]) hi ha el registre d'estat associat que ens permet realitzar la comunicació utilitzant el protocol de handshaking. La lectura del port 3 produeix com efecte lateral la posta a zero del port 4. El senyal Alarma, d'un sol bit, anirà connectat al bit de major pes del port 5 de les sortides (OUTPUT[5]). Els altres 15 bits d'aquest port els hem de posar a 0. El dispositiu d'alarma té com a entrada el bit de més pes d'OUTPUT[5]. Quan aquest està a 1, sona l'alarma, i quan està a zero no sona, per tant no cal utilitzar cap protocol de comunicació amb el dispositiu d'alarma.

Dibuixeu el graf d'estats de l'UCE (no oblideu la llegenda) perquè el PPE realitzi la funcionalitat especificada. Indiqueu en cada estat la paraula de control amb mnemotècnics.



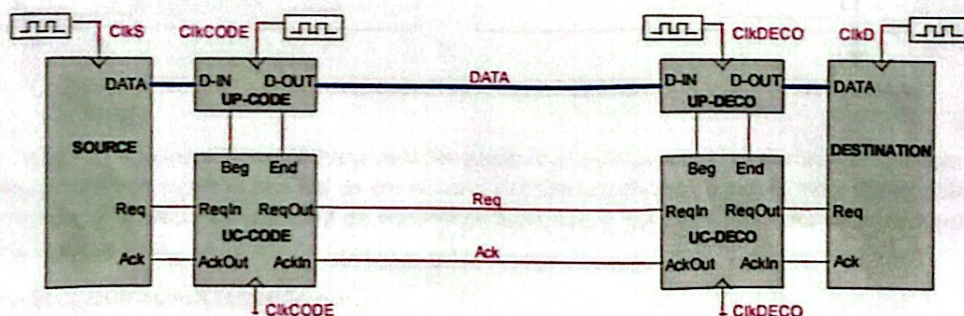
Exercici 5

La siguiente figura muestra un sistema para la transmisión de datos a larga distancia desde un emisor (SOURCE) hasta un receptor (DESTINATION) mediante un bus de datos (DATA) y las señales necesarias para implementar el protocolo de comunicación asincrónica de cuatro pasos: Req (del emisor al receptor, ya que es el emisor quien inicia la transmisión de cada dato) y Ack (del receptor al emisor). (Observad que, como es lógico, el reloj del emisor (ClkS) y el del receptor (ClkD) son distintos).



Ante la sospecha de que el bus de datos (DATA) y las señales Req y Ack están siendo pinchados en algún punto del recorrido entre el emisor y el receptor y la posibilidad de que alguien no autorizado pueda acceder a la información transmitida, se decide encriptar cada dato a la salida del emisor y descryptarlo a la llegada al receptor, para que la información que viaje por el largo bus sea irreconocible por alguien no autorizado.

El sistema final se muestra en la siguiente figura. Se ha cortado el bus DATA y las señales Req y Ack cerca del emisor y se ha insertado allí el encriptador (CODE) para que transforme las señales adecuadamente. El encriptador está formado por una unidad de proceso (UP-CODE) y una unidad de control (UC-CODE). El descryptador (DECO = UP-DECO + UC-DECO) hace lo propio cerca del receptor. El objetivo es que SOURCE y DESTINATION se comuniquen con el protocolo asincrónico como lo hacían antes de insertar CODE y DECO entre ellos. (Observad que los relojes del encriptador y descryptador (ClkCODE y ClkDECO) son distintos entre sí y distintos a los demás relojes).



El descryptador ya está diseñado. Veamos el funcionamiento del encriptador: primero de su unidad de proceso, que ya está diseñada, y después de su unidad de control, de la que tendréis que dibujar el grafo de estados.

UP-CODE. La unidad de proceso inicia el encriptado del dato presente en su bus de entrada D-IN cada vez que su señal de entrada Beg es puesta a 1 durante un ciclo (de su reloj ClkCODE). El número de ciclos del reloj que tarda en encriptar un dato depende del valor concreto del dato y siempre es mayor que uno. Durante todos los ciclos que dura la encriptación el dato en el bus D-IN debe mantenerse fijo. Cuando la unidad de proceso ha terminado de encriptar el dato presente en D-IN copia el dato encriptado en el registro de salida del bus D-OUT y cuando el dato encriptado está estable en D-OUT activa durante un ciclo la señal End. El dato encriptado permanece fijo en el bus D-OUT hasta que se termine de encriptar el siguiente dato. Si durante la encriptación de un dato se activa la señal Beg la unidad de proceso aborta la encriptación actual e inicia la encriptación del dato que se encuentre en ese ciclo en D-IN.

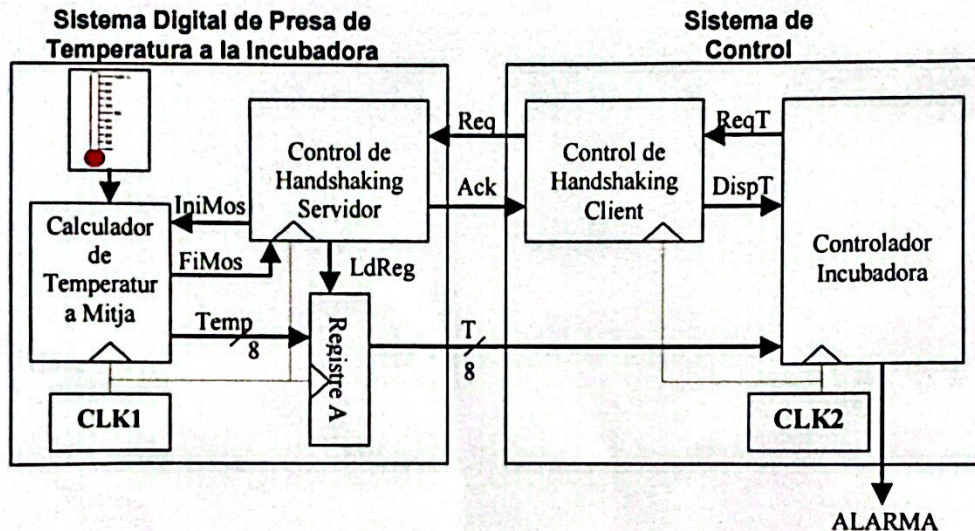
UC-CODE. Este bloque controla a la unidad de proceso para que encripte cada dato correctamente. Además, gestiona las señales del protocolo asincrónico para que SOURCE y DESTINATION se comuniquen igual que lo hacían en el sistema original, en el que no existían ni CODE ni DECO. La UC no debe modificar la señal ReqOut hasta que no recibe la modificación correspondiente por ReqIn. Lo mismo debe ocurrir entre AckOut y AckIn. Por último, suponed que cuando se inicializa el sistema las señales Req y Ack valen 0 y que tanto SOURCE como DESTINATION realizan el protocolo de handshaking de 4 pasos correctamente y que DECO también realiza la descryptación y mantiene la transparencia del protocolo correctamente: por ello UC-CODE no debe considerar la posibilidad de que haya un error en el protocolo de 4 pasos.

Dibujad el grafo de Moore del circuito secuencial UC-CODE. No olvidéis incluir la leyenda del grafo ya que si no hay leyenda o esta no es correcta la solución es totalmente incorrecta.

Exercici 6

Davant de la creixent demanda d'aletes de pollastre per influència nord-americana, una empresa pionera en el sector ha descobert un mètode per modificar genèticament els pollastres aconseguint que neixin amb quatre ales. Per disminuir la probabilitat de malformacions als pollastres, cal controlar l'evolució de la temperatura a la que es troben els ous a la incubadora. Cada incubadora porta incorporat un Sistema Digital de Presa de la Temperatura governat per un senyal de clock propi de freqüència desconeguda. Aquest sistema, tal com es mostra a la figura, té una entrada Req, que li indica que es vol obtenir la temperatura mitjana d'una mostra, i una sortida Ack, que indica que s'ha rebut la petició i que la temperatura demandada ja està disponible al bus de dades de 8 bits T.

L'empresa ha demanat a un professor d'IC que dissenyi el Sistema de Control del circuit encarregat de supervisar el procés d'incubació dels pollastres de quatre ales, que activarà una alarma en cas que es detecti una anomalia en el procés. L'esquema del circuit proposat pel professor és el següent:



Per tal d'aconseguir que el Sistema Digital de Presa de la Temperatura a la Incubadora ens retorni la temperatura mitjana, els dos sistemes es comuniquen mitjançant el protocol de comunicació Handshaking de quatre passes (Four-Cycle), utilitzant **Control de Handshaking Client**, que implementa el control de protocol de handshaking del client, i **Control de Handshaking Servidor**, que implementa la part del servidor.

A més afegeix unes quantes notes a l'especificació:

- Quan **Controlador Incubadora** necessita obtenir la temperatura mitja d'un interval de temps, posa el senyal **ReqT** a 1 durant un cicle del clock CLK2.
- Quan el resultat està disponible al bus T, **DispT** s'ha de posar a 1 durant un únic cicle del rellotge CLK2. Si es manté a 1 més d'un cicle, **Controlador Incubadora** podria funcionar incorrectament.
- Una vegada **Controlador Incubadora** ha fet una petició posant **ReqT** a 1, no en pot fer una altra fins al cicle següent de rebre el senyal **DispT**.
- Si **Controlador Incubadora** ja ha rebut el senyal **DispT** i decideix iniciar una altra consulta abans que **Control de Handshaking Client** estigui preparat per iniciar una nova transmissió, cal que aquest últim tingui en compte que hi ha una petició pendent que ha d'iniciar tan aviat com sigui possible. (No es pot acumular més d'una petició)
- Quan s'inicialitzen els sistemes, es garanteix que el senyal **Ack** val 0.

El professor, quan només li faltava especificar el disseny de **Control de Handshaking Client**, se n'ha adonat que el que estava fent als pollastres no era gaire ètic i ho ha deixat a mitges. L'empresa ha decidit que un estudiant sense escrúpols acabi la feina:

- a) Especifica la llegenda del graf de Moore que descriu el comportament del circuit lògic seqüencial **Control de Handshaking Client**.
- b) Fes el graf de Moore del bloc **Control de Handshaking Client**. Recorda que el comportament del sistema ha d'estar definit a cada estat per totes les possibles combinacions dels valors d'entrada.