

Desarrollo de software para sistemas empotrados

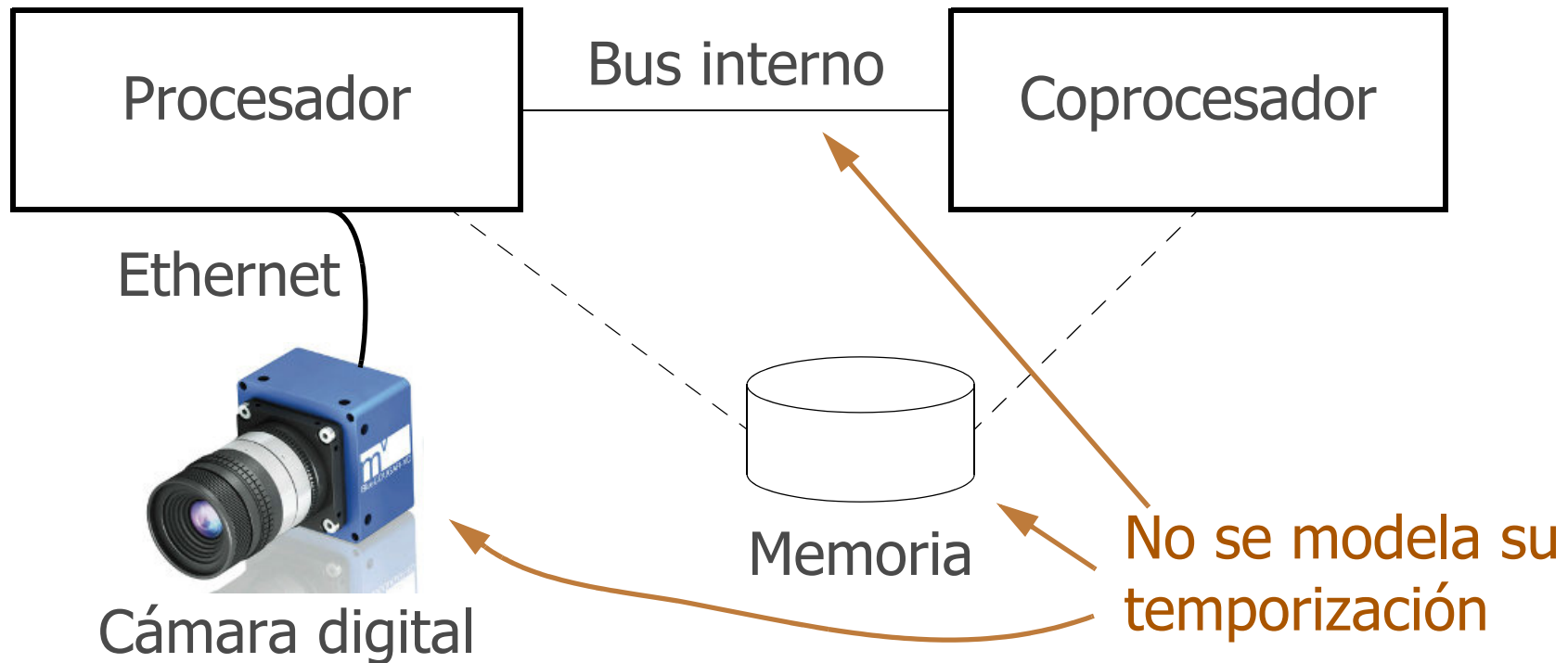
Examen Febrero 2017

Introducción

El objetivo de este ejercicio es desarrollar parte de un sistema de control de un sistema de visión artificial compuesto por una cámara digital, un procesador y un coprocesador de imágenes

- la cámara se supone que toma la imagen de manera instantánea y a partir de ese momento la envía por una red Ethernet
 - la temporización de la cámara no es relevante y por ello *no* se modela
 - sin embargo la transmisión de la imagen por Ethernet *sí* se modela
- el procesador y coprocesador comparten una memoria para intercambiar imágenes y datos
 - esta memoria no influye en la temporización y *no* se modela

Plataforma



El *bus interno* conecta el procesador y el coprocesador, no influye en la temporización y esta *no* se modela

La *red Ethernet* de 1Gbit/s se usa para que la cámara envíe imágenes de forma periódica, cada 10 ms

Software funcional

La funcionalidad del software está ya desarrollada en forma de 3 funciones

- dos de ellas para funcionar en el procesador
 - `procesamiento_imagen()`: hace el procesamiento de la imagen recibida de la cámara; en mitad de su ejecución envía un trabajo al coprocesador y se suspende en espera de la respuesta
 - `historico()`: gestiona un registro histórico de las imágenes
- una para el coprocesador
 - `seguimiento()`: recibe un puntero a una imagen guardada en memoria, hace un seguimiento de un objeto que aparece en la imagen, y retorna datos con la posición del objeto

Requisitos funcionales

1. La cámara envía una imagen por la red Ethernet cada 10 ms
 - la imagen ocupa 3 frames de Ethernet de 8Kbytes cada uno
2. El procesador invoca `procesamiento_imagen()` cada vez que recibe una imagen
 - como se ha mencionado, a mitad de su ejecución manda una petición al coprocesador, se suspende en espera de la respuesta y luego continúa su ejecución
3. Cada vez que el coprocesador recibe una petición invoca la función `seguimiento()` y retorna la respuesta
4. El procesador invoca periódicamente la función `historico()`

Requisitos temporales (no funcionales)

1. Los tiempos de ejecución de peor caso medidos para las funciones software ya disponibles son los siguientes:
 - `procesamiento_imagen()`: 4 ms (dividida en dos partes de 2ms)
 - `seguimiento()`: 5 ms
 - `historico()`: 43 ms
2. La cámara manda una imagen cada 10ms. Su procesamiento debe estar terminado en un plazo de 10ms
3. El histórico se hace cada 100ms. Esta actividad debe estar terminada en un plazo de 100ms

Estos requisitos deben validarse con un análisis de planificabilidad

Otros requisitos no funcionales

Los desarrollos y el software básico estarán basados en la plataforma indicada arriba, con un procesador convencional con un sistema operativo gobernado por eventos, con prioridades fijas

- el rango de prioridades es 1..32 en el procesador
 - el nivel 32 es el nivel de interrupción
- el coprocesador funciona como un procesador independiente, y se puede suponer que tiene un planificador por prioridades ya que solo se ejecuta una actividad
 - la prioridad asignada es indiferente
- el tiempo de cambio de contexto en el procesador es de $0.5 \mu\text{s}$
- el sistema operativo tiene un temporizador de tipo `Alarm_Clock`, con un tiempo de "overhead" de $0.3 \mu\text{s}$

Ejercicios

1. Dentro del proceso de análisis de requisitos, generar diagramas UCM para los requisitos del sistema
 - exceptuando los de la plataforma
2. Modelar con AADL una arquitectura para este sistema, con dos flujos de eventos:
 - a) procesamiento de imágenes, con:
 - la cámara, capaz de enviar mensajes de forma periódica
 - el mensaje con la imagen producida por la cámara
 - un thread para el procesamiento de imágenes, activado por la llegada de la imagen
 - un thread para el seguimiento, activado por la petición que procede del procesador
 - b) Registro del histórico de imágenes, con:
 - un thread periódico para el registro del histórico

Ejercicios (cont.)

3. Modelar con MAST la arquitectura del sistema para realizar un análisis de planificabilidad inicial
- ¿Cuál es la asignación óptima de prioridades?
 - ¿Es planificable el sistema con esa asignación?
 - ¿Cuánto podríamos aumentar los tiempos de ejecución y seguir manteniendo el sistema planificable?
 - en el procesador
 - en el coprocesador

Entregar

Un informe en pdf con:

- diagrama(s) UCM
- diagrama AADL de nivel de sistema
- una captura de pantalla de los resultados de MAST y las respuestas a las preguntas planteadas

Workspace de UCMNav comprimido

Workspace de OSATE comprimido

Ficheros del modelo MAST en un archivo comprimido