

## TPS ARQUITECTURA DE COMPUTADORAS

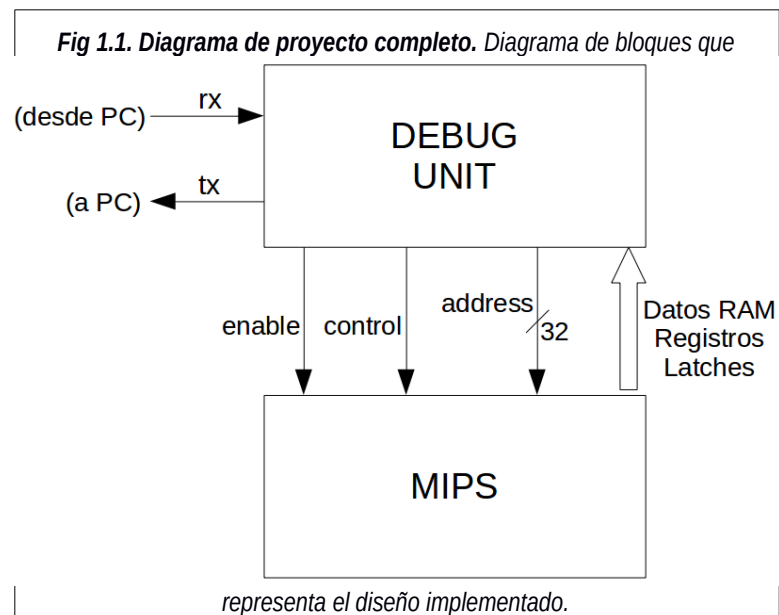
Nazzi, Fabio

Saavedra, Leandro

Sagripanti, Sergio

## 1. Introducción

Se realiza la implementación de una versión simplificada del procesador MIPS y un módulo para debugging (Fig 1.1).



## 2. Implementación MIPS

## 3. Riesgos

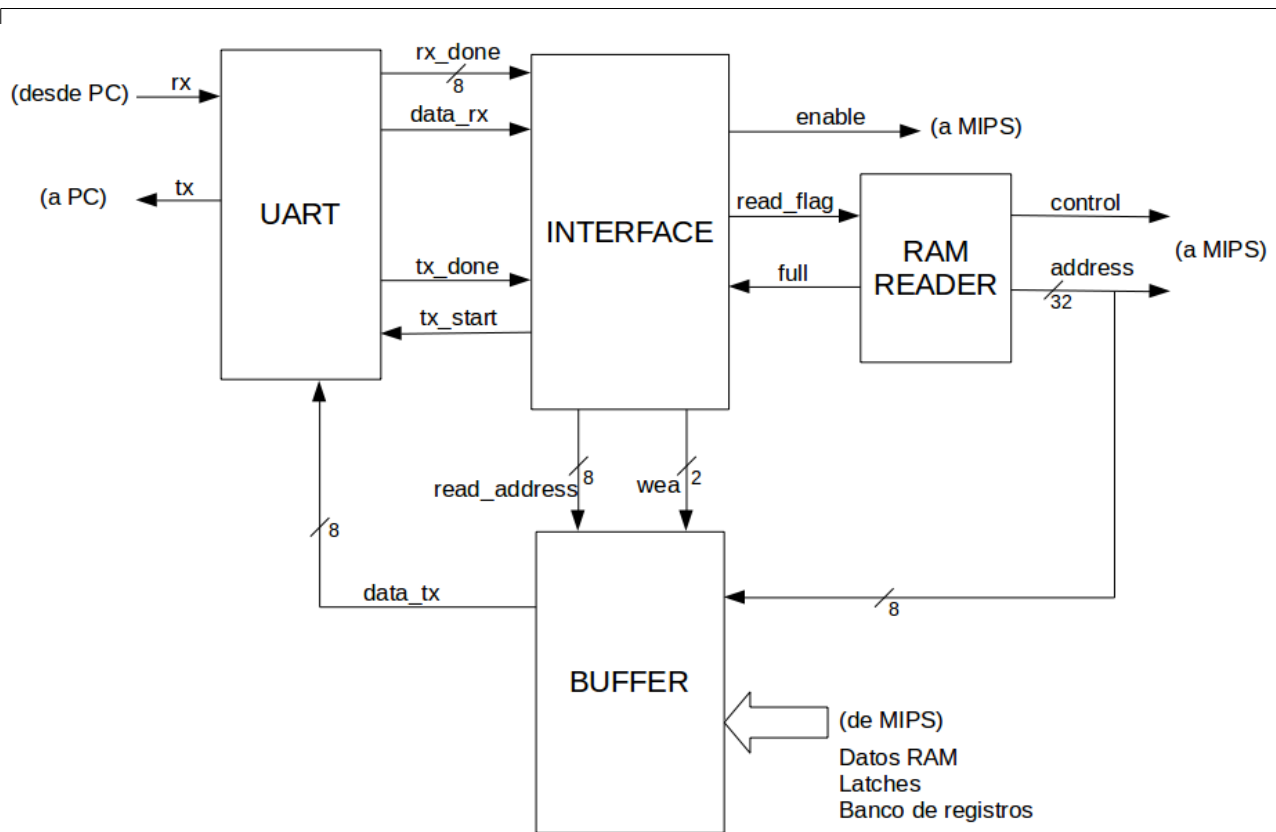
Riesgos a resolver  
Cómo se resolvieron

## 4. Debug Unit

La unidad de debugging (Fig 4.1) esta compuesta por cuatro módulos:

- UART
- Buffer
- Interface
- RAM\_Reader

A continuación, se explicará en detalle cada uno de ellos.



**Fig 4.1. Módulo Debugger.** Diagrama de bloques que muestra los componentes internos y su conexión

## 4.1 UART

El módulo UART (Fig. 4.2) se encarga de la comunicación entre el módulo MIPS y la computadora.<sup>1</sup>

### 4.1.1 Funcionamiento

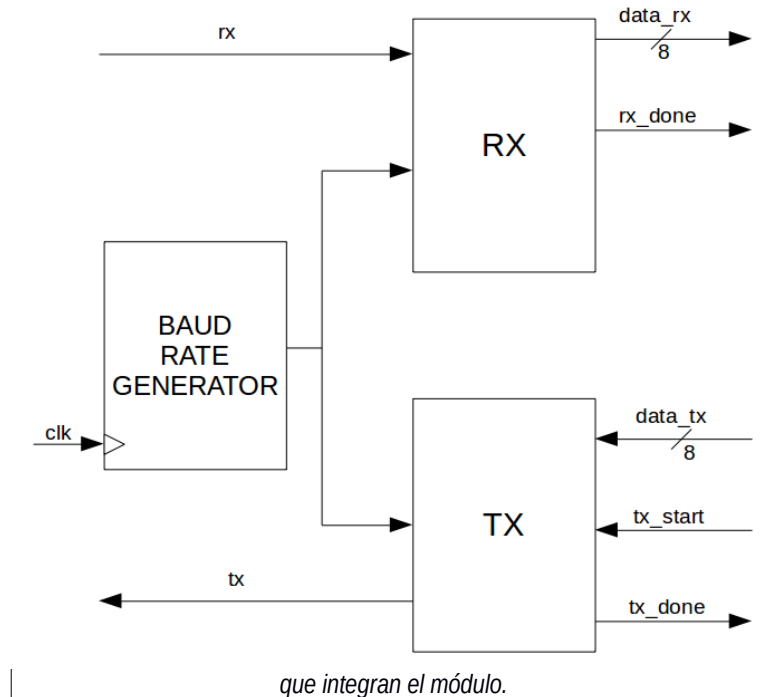
El componente *Baud Rate Generator* está constituido por un contador módulo 651. Genera una señal durante un ciclo de clock al llegar a este valor, permitiendo al módulo UART trabajar a una tasa de 9600 Baudios. Dicha señal es usada por los componentes *RX* y *TX*.

Cuando el módulo *RX* detecta un bit de start en el bus *rx*, comienza la recepción de datos. Cuando el dato completo es adquirido, lo coloca en el bus de salida *data\_rx*. Finalmente, activa una señal *rx\_done*, para notificar que el dato está disponible para ser usado.

Por su parte, el módulo *TX*, mientras no tiene datos para enviar, mantiene la línea de salida *tx* a un valor lógico 1. Cuando la señal de entrada *tx\_start* se activa, toma el valor del bus de entrada *data\_tx* para su posterior envío. Una vez enviado el dato, activa la señal *tx\_done* para notificar la operación, durante un ciclo de clock.

<sup>1</sup> Para mas detalles de diseño e implementación ver informe *UART.docx* adjunto.

**Fig 4.2. UART.** Diagrama de bloques que muestra los componentes internos



## 4.2 Buffer

El módulo Buffer contiene un arreglo de registros que proporciona almacenamiento para los datos de la memoria RAM. Además, proporciona el mecanismo de selección de los datos a ser enviados por el modulo UART.

### 4.2.1 Funcionamiento

El funcionamiento de este módulo se divide en dos etapas. La señal de entrada *wea*, controlada por el módulo Interface, le indica el modo de operación.

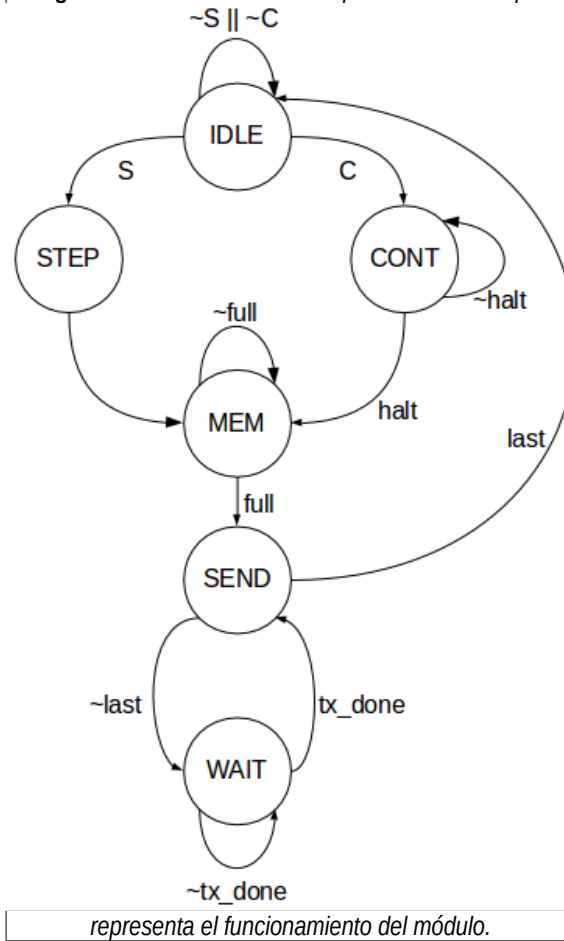
En primer instancia, el módulo almacena los datos provenientes de la memoria RAM del procesador en un arreglo de registros. Cada dato es ubicado en la posición apuntada por la señal *address* (8 bits de menos peso) conectada a la salida del módulo RAM\_Reader.

La segunda etapa utiliza la señal *read\_address* para seleccionar los datos (de 8 bits) a ser enviados por el módulo UART. Los datos a enviar son los de la memoria RAM, almacenados en la etapa previa, y el contenido de registros y latches asignados directamente a la salida desde el procesador MIPS. Cada dato es asignado al bus de salida *data\_tx*.

## 4.3 Interface

Este módulo es el encargado de gestionar y coordinar el funcionamiento del procesador, delegar la lectura de la RAM al modulo RAM\_Reader y controlar su operación, y coordinar la comunicación a través del modulo UART. Fue diseñado e implementado como una máquina de estados finita (Fig 4.3)

**Fig 4.3. Módulo Interface.** Máquina de estados que



### 4.3.1 Funcionamiento

La unidad Interface controla el funcionamiento del procesador, permitiéndole funcionar en dos modos: paso a paso y continuo.

Inicialmente, se encuentra a la espera de un dato proveniente del módulo UART a través del bus *data\_rx*. Si dicho dato no corresponde con el valor ASCII asociado a las letras S o C, permanece en el mismo estado. Por el contrario, si el caracter recibido corresponde con alguno de los valores antes mencionados, se determina uno de los modos de operación del procesador.

El modo de funcionamiento paso a paso se consigue si el caracter recibido corresponde a una S. En este caso, activa la señal de salida *enable* durante un ciclo de clock, que permite que el procesador esté operativo durante ese período.

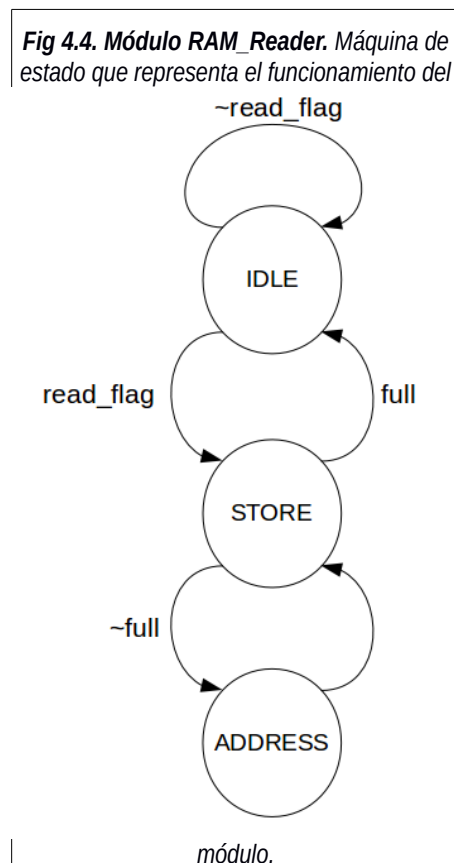
Luego, delega al módulo RAM\_Reader la lectura de la memoria de datos del procesador a través de la señal de control *read\_flag*. Al mismo tiempo, utiliza la señal *wea* conectada a la unidad Buffer para indicarle que debe almacenar los datos leídos.

Finalizada la operación anterior y notificada mediante la señal *full*, manejada por RAM\_Reader, el módulo Interface comienza a controlar el envío de los datos proporcionados por Buffer. Selecciona cada byte a enviar mediante el bus *read\_address*. La transmisión llevada a cabo por la unidad UART se activa a través de la señal *tx\_start*, mientras que el fin de la operación es notificado con la señal *tx\_done*.

Si el carácter recibido del módulo UART corresponde a un valor ASCII igual a C, se determina el modo de operación continuo del procesador. En este modo, la señal *enable* permanece activa, permitiéndole al procesador ejecutar todas las instrucciones almacenadas en la memoria ROM. Es éste quien notifica, mediante la señal *halt*, que ha terminado la ejecución, dando lugar a que el módulo Interface continúe su operación. Luego, éste procede de igual manera que en el caso anterior.

#### 4.4 RAM\_Reader

A esta unidad se le delega la función de leer los datos de la memoria RAM y pasarlos al módulo Buffer para ser almacenados.



##### 4.4.1 Funcionamiento

El módulo RAM\_Reader (Fig 4.4) espera que la señal *read\_flag*, manejada por el módulo Interface, se active. Esto cambia el estado de la unidad, activando la señal de salida *control* que le permite acceder a la memoria RAM del procesador desde el exterior.

El bus de salida *address* direcciona la memoria RAM para acceder a los datos almacenados en ella.

Finalizada la lectura de los datos, el módulo vuelve al estado inicial reiniciando sus valores internos.