

Facultad de Ingeniería - UdeLaR

# Estudio estructural del hardware del equipo Sun SPARCcenter 2000

Gerardo Ares Meneces

Copyright © 2002 gares@fing.edu.uy

Last Revision Date: 19 de noviembre de 2002

# Table of Contents

1. Arquitectura del sistema
  - 1.1. Características generales
  - 1.2. Lógica del sistema
  - 1.3. System Board
  - 1.4. Componentes principales
2. XDBus
  - 2.1. Protocolo del Bus
  - 2.2. XDBus en ambiente multiprocesador
  - 2.3. Esquema para arbitraje del bus
3. Unidad de Procesador
  - 3.1. Módulo CPU
    - Cache de 1er. nivel y 2do. nivel
  - 3.2. Interconexión CPU-XDBus
4. Unidad de Memoria
5. Unidad de Entrada/Salida
6. SPARC V8
  - 6.1. Modelos de memoria
    - TSO ● PSO

**6.2. Reference Memory Managment Unit**

- Traducción de direcciones

**6.3. Soporte para Multiprocesador**

## Índice de figuras

|     |   |    |
|-----|---|----|
| 1.  | Arquitectura del sistema . . . . .                      | 6  |
| 2.  | Esquema general . . . . .                               | 7  |
| 3.  | Board . . . . .   | 8  |
| 4.  | XDBus . . . . .   | 10 |
| 5.  | Packet-Switched Bus . . . . .                           | 11 |
| 6.  | Pipelined . . . . .                                     | 13 |
| 7.  | Arquitectura de arbitraje . . . . .                     | 17 |
| 8.  | Unidad de Procesador . . . . .                          | 18 |
| 9.  | Unidad de memoria . . . . .                             | 22 |
| 10. | Unidad de Entrada/Salida . . . . .                      | 23 |
| 11. | Modelo TSO . . . . .                                    | 26 |
| 12. | Modelo PSO . . . . .                                    | 27 |
| 13. | Estandarización para el acceso a memoria . . . . .      | 28 |
| 14. | Dirección física y dirección virtual . . . . .          | 29 |
| 15. | Traducción de direcciones virtuales a físicas . . . . . | 30 |

# 1. Arquitectura del sistema

## 1.1. Características generales

- SPARC (Scalable Processor ARChitecture) - Arquitectura RISC
- SPARC Version 8 - Microprocesadores SuperSPARC II
- Direcccionamiento de 32 bytes
- Arquitectura de Bus Multinivel
- Clasificación
  - MIMD Multiple Instruction Multiple Data
  - Tightly coupled system
- Diseño Modular
  - Unidad Procesador
  - Unidad Memoria
  - Unidad Entrada/Salida

## 1.2. Lógica del sistema

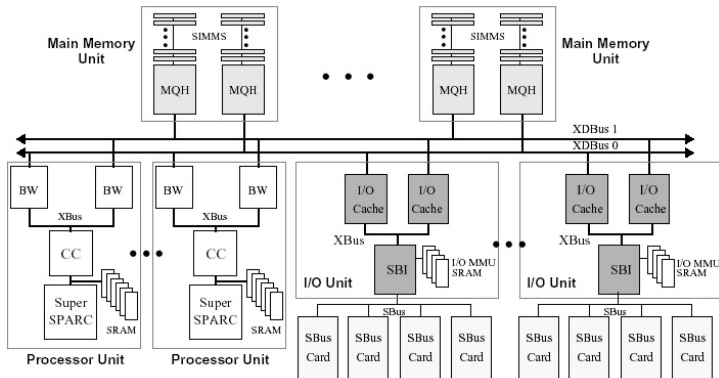


Figura 1: Arquitectura del sistema

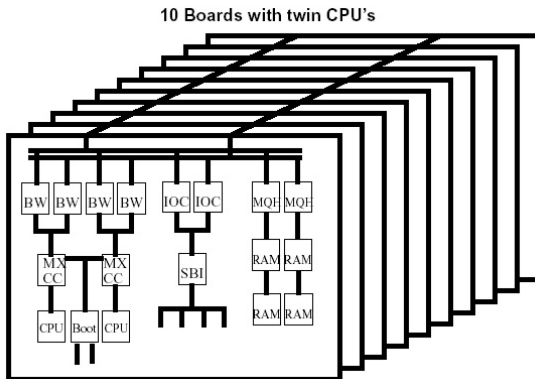


Figura 2: Esquema general

## 1.3. System Board

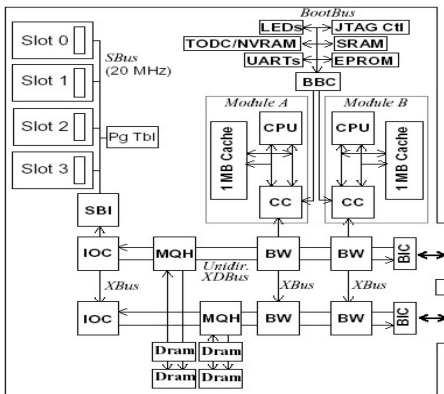


Figura 3: Board



## 1.4. Componentes principales

- Control Board
- 10 System Boards
- 20 procesadores SuperSPARC - 2 módulos por System Board
- Memoria RAM 5GB - 512 MB por Board
- 40 dispositivos de Entrada/Salida - 4 por Board
- 2 MB SuperCache por CPU
- Arquitectura: Multilevel Bus
  - 2 XDBus
  - XBus
  - SBus
- NVRAM

## 2. XDBus

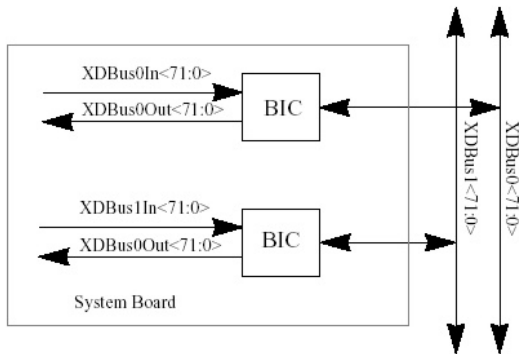


Figura 4: XDBus

- Packet-Switched bus (Request, Reply)
- Sincrónico

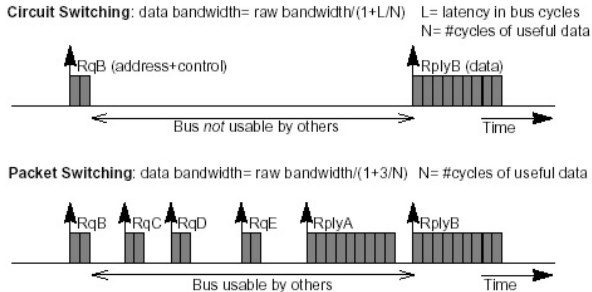


Figura 5: Packet-Switched Bus

- High-Bandwidth (320MB/s-400MB/s)
- High-Performance (75 %)
- BIC (Bus Interface Chips) 16 datos + 2 paridad
- Protocolo para coherencia de cache en ambiente de multiprocesadores
- 88 señales
  - 72 control de datos (64 datos + 8 paridad)
  - 13 comunicación con arbitro (point-to-point)
  - 3 clock y otros de control
- Propósito general (chips, boards, backplane)
- Prioridad
  - Paquetes de *reply* tienen mayor prioridad

- Pipelined (Segmentado)
  - Modo unidireccional

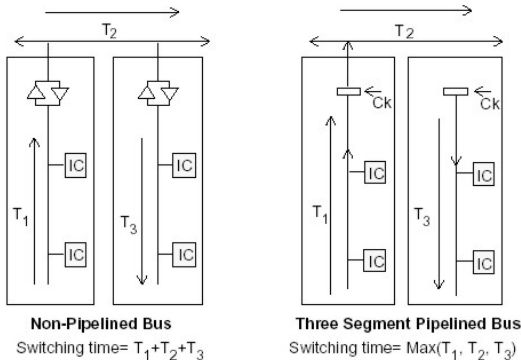


Figura 6: Pipelined

## 2.1. Protocolo del Bus

- Ciclos (eléctrica)
  - Unidad de tiempo: 1 clock del bus (40/50 Mhz)
- Paquetes (lógica)
  - Secuencia contigua de ciclos
  - Paquetes
    - Request: 2 ciclos
    - Reply: 9 ciclos
  - No preemptivo
- Transacciones
  - Conseguir el 'bus mastership' (lineas dedicadas)
  - Transmitir el 'request' (bus de datos)
  - Conseguir el 'bus mastership' (lineas dedicadas)
  - Transmitir el 'reply' (bus de datos)

## 2.2. XDBus en ambiente multiprocesador

Coherencia de cache (Cache coherency) Generalización del protocolo 'multy-copy write-broadcast'

- Adaptar el protocolo al 'packet switched bus'
  - Problema: No atómica la transacción
  - Solución: Tratar un *read* como si fuera hecho en el paquete *request*. Tratar un *write* como si fuera hecho en el paquete *reply*
- Permitir al hardware emular esquemas de coherencia
  - *write-invalidate*: datos poco compartidos
  - *write-update*: datos muy compartidos
- Soporte para coherencia de cache para una jerarquía multinivel de cache
  - Provee de localización de tráfico de datos
  - Escalabilidad a cientos de procesadores

- Operación *DeMap* que permite hacer un 'flush' sincrónico a todas las tablas TLB (Translation Look-aside Buffer)
- 2 Transacciones para permitir atomicidad:
  - Lock y Unlock
- Manejo de interrupciones
  - Transacción para transportar una interrupción desde un dispositivo o procesador a uno o más procesadores
  - Soporte para programar el dispositivo para que interumpa determinado procesador, sino interrupción dinámica
    - Dispositivo genera interrupción
    - Se determina el procesador (sino esta seteado) (CARB)
    - Se envía un paquete al XDBus para el procesador a interrumpir
    - BusWatch detecta y pasa el mensaje al SuperCache Controller, luego al procesador



## 2.3. Esquema para arbitraje del bus

- Jerárquico
  - Arbitro central (Control Board - CARB)
  - Board Bus Arbiters (System Board - BARB)

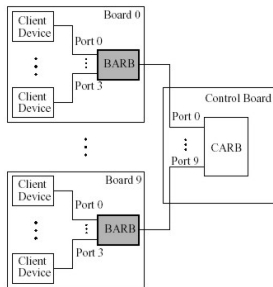


Figura 7: Arquitectura de arbitraje

### 3. Unidad de Procesador

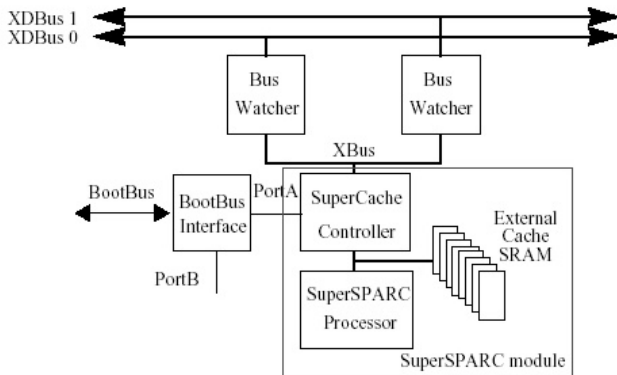


Figura 8: Unidad de Procesador

### 3.1. Módulo CPU

- Microprocesador SuperSPARC II 85 MHz (Superescalable)
  - 3.1 M transistores
  - Unidad de enteros (UI)
  - Unidad de punto flotante (FPU)
  - Pipeline de 4 ciclos (Fetch, Decode, Execute y Write-Back)
  - On-chip cache (First level cache)
    - Cache de instrucciones (20 kb, 5-way set-associative)
    - Cache de datos (16 kb, 4-way set-associative)
  - Unidad para manejo de referencias a memoria (RMMU)
  - Interconexión por MBUS o XBus
- Controlador de cache externo (Second level cache)
  - 2.2 M transistores
  - Soporta hasta 2 MB (Direct-mapped)

- **Cache de 1er. nivel y 2do. nivel**

| Cache type             | Size./Org.   | Protocol                      | Block Size/Line Size |
|------------------------|--------------|-------------------------------|----------------------|
| Instruction, 1st level | 20 kb, 5-way | invalidate, invalidate        | 32B/64B              |
| Data, 1st level        | 16 kb, 4-way | write-through, invalidate     | 32B/32B              |
| Combined, 2do level    | 2 MB         | write-back, update/invalidate | 64B/256B             |

- Cache de 1er. nivel (First level cache)
  - n-way set-associative (address tag)
  - LRU Algorithm (*least-recently-used*)
- Cache de 2do.nivel (Second level cache)
  - Direct-map, con cache de datos e instrucciones
  - Soporte para *block copy* y *block zero*

## 3.2. Interconexión CPU-XDBus

- Bus Watcher
  - Permite los algoritmos de coherencia de memoria
  - Mejora la utilización del XDBus, baja la contención
  - Cache tags replicadas de CC
  - Espía el XDBus (*Snooping*)
  - Filtra el tráfico que lee del XDBus
- XBus (*packet-switch bus*)
  - Comunicación entre el BW y el CC
  - Mismas características que el XDBus + operaciones para manejo de información privada (cache tags)
  - Arbitro implementado en CC (mejor eficiencia)
  - Dispositivos identificados con un *XBusId*

## 4. Unidad de Memoria

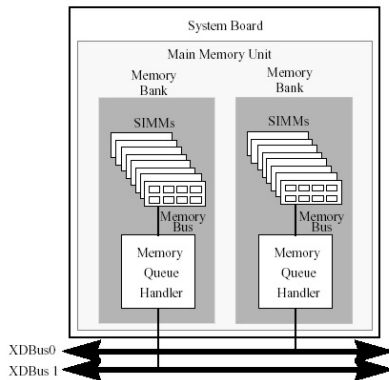


Figura 9: Unidad de memoria

## 5. Unidad de Entrada/Salida

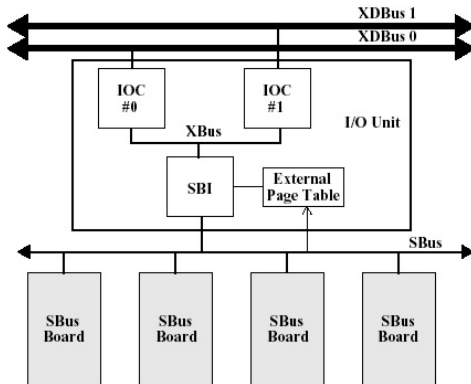


Figura 10: Unidad de Entrada/Salida

- Tres modelos para acceso a los dispositivos:
  - *Programmed I/O* el procesador lee y escribe directamente del dispositivo
  - *Consistent DVMA I/O* (Direct Virtual Memory Access)  
Los datos son transferidos directamente desde el dispositivo a la memoria. Se utiliza la *tabla de páginas externa* para la traducción de direcciones de SBus a XDBus. Se utiliza el IOC.
  - *Stream Mode DVMA* Los datos no son transmitidos directamente, sino que pasan por un par de *buffers* que se encuentran en el SBI. Es manejado por software. Las traducciones también son realizadas por el XPT
- La tabla de páginas externa puede mapear hasta 64MB de DVMA. El kernel debe de proveer la consistencia de la tabla.
- SBus
  - 20 Mhz
  - Soportados por varias interfaces



## 6. SPARC V8

### 6.1. Modelos de memoria

- Define la semántica de las operaciones en memoria (*load* y *store*)
- Especifica como el orden en el cual las operaciones son ejecutadas por un procesador estan relacionadas con el orden en el cual son ejecutadas por la memoria
- Modelos:
  - TSO *Total Store Ordering*
  - PSO *Partial Store Ordering*
- N puertos de acceso ( $n$  procesadores)
- Cada procesador tiene un *store-buffer*
- Single-port memory (switch)

## • TSO

- El orden de la secuencia de operaciones *store*, *flush* y *atomic load-store* hechas en el procesador es idéntica al orden hecho en memoria
- El *load* es buscado primero en el *store-buffer* y luego en memoria (bloquea el procesador)

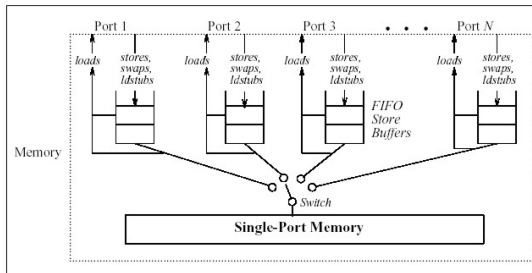


Figura 11: Modelo TSO

- **PSO**

- El orden de la secuencia de operaciones *store*, *flush* y *atomic load-store* hechas en el procesador NO es necesariamente idéntica al orden hecho en memoria
- Instrucción *STBAR* (Divide al *store-buffer*)

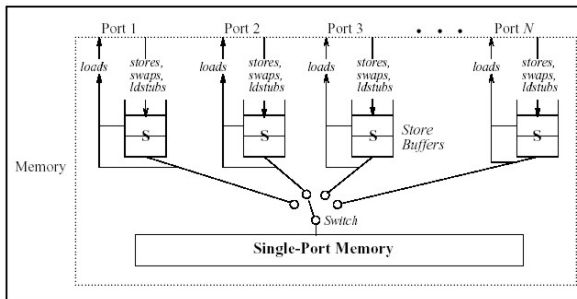


Figura 12: Modelo PSO

## 6.2. Reference Memory Managment Unit

### ■ Funcionalidades:

- Realizar la traducción de direcciones virtuales a físicas
- Protección de la memoria - Control que los procesos no puedan leer o escribir en el espacio de direccionamiento de otro proceso
- Implementa memoria virtual - Lleva cuenta de que páginas estan en memoria y por lo tanto genera *page fault* cuando es necesario

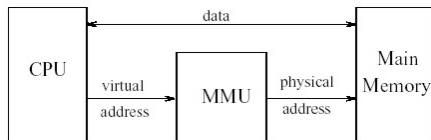


Figura 13: Estandarización para el acceso a memoria

- **Traducción de direcciones**

- Traduce direcciones virtuales de 32-bits a direcciones físicas de 36-bits
- La memoria física es dividida en páginas de 4KB
- 20 bits son utilizados para la traducción de la página física y 12 para *offset* dentro de la página
- Permite un nivel de direccionamiento de 64GB

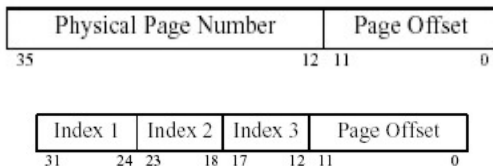


Figura 14: Dirección física y dirección virtual

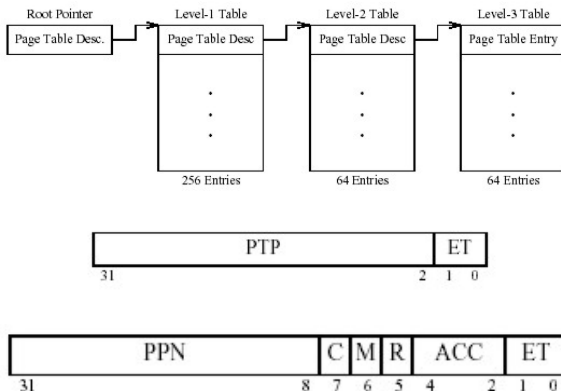


Figura 15: Traducción de direcciones virtuales a físicas

### 6.3. Soporte para Multiprocesador

La arquitectura SPARC brinda 2 operaciones atómicas especializadas para ambientes multiprocesadores

- *swap* Intercambia el contenido de un registro (UI) con una palabra en memoria
- *ldstwb* (load and store unsigned byte) Lee un byte de la memoria a un registro y sobrescribe en memoria el mismo byte con 1s (unos)

Son eficientes para mecanismos de sincronización como semáforos

## Referencias

- [1] *Multiprocessor System Architectures*. Ben Catanzaro, 1994.
- [2] *SPARCcenter 2000: Multiprocessing for the 90's!*. Michel Cekleov. IEEE COMPCON 1993.
- [3] *The Next-generation SPARC Multiprocessing System Architecture*. Jean-Marc Frailong. IEEE COMPCON 1993.
- [4] *The XDBus: A High Performance, Consistent, Packet Switched VLSI Bus*. Pradeep Sindhu. IEEE COMPCON 1993.
- [5] *The SPARC Architecture Manual (version 8)*. Prentice Hall, 1992.
- [6] *SPARCcenter 2000E & SPARCserver 1000E as Compute Application Servers*. Sun Microsystems, 1995.
- [7] *The SuperSPARC-II Microprocessor Architecture*. Sun Microsystems, 1995.
- [8] *Computer Architecture a Quantitative Approach*. Hennessy & Patterson, 2nd.Edition, 1996.