

# S11-Sistemas de Entrada/Salida

Andre Rosero, Ricardo León, Mauro Valencia, David Paredes, Adrián Trujillo, Juan Jácome, Nayeli Merchan, Enrique Betancourt, Samara Salazar

January 18, 2026

- Sistemas de Entrada/Salida (E1,11,270)(E1, 7,229)
- Módulos de E/S (E2, 11,273)(E2, 7,236)
- E/S Programada (E4,11,276)(E4,7,239)
- Instrucciones de E/S (E4,11,278)(E4,7,241)
- E/S mediante Interrupción (E3,11,280)(E3,7,243)
- Acceso directo a Memoria DMA (E5,11,290)(E5,7,251)

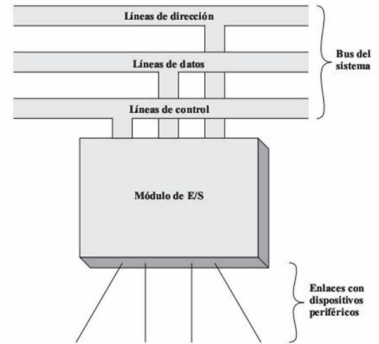
**EL MÓDULO DE E/S** Esta sección explica la arquitectura: cómo se conecta el computador con el mundo. El Módulo de E/S y su Función

- **Definición:** Es el tercer elemento clave del computador (junto a CPU y Memoria). No es solo un conector: Un módulo de E/S no solo permite enchufar el dispositivo al bus de sistema; tiene cierta “inteligencia”, para permitir la comunicación entre el periférico y el bus
- **Función principal:** Interfaz con el procesador y memoria (vía Bus del Sistema). Interfaz con uno o más dispositivos periféricos (vía enlaces de datos).

¿Por qué no conectar periféricos directamente?

- **Diversidad:** Existen demasiados tipos de periféricos con métodos de operación distintos.
- **Velocidad:** La mayoría son mucho más lentos que la CPU/RAM. Amenudo la velocidad de transferencia de datos de los periféricos es mucho menor que la de la memoria o el procesador.
- **Formatos de Datos:** Los periféricos usan formatos y tamaños de palabra diferentes a los del computador.
- **Conclusión:** El Módulo de E/S actúa como traductor y regulador de velocidad.

- 1 **E/S Programada:** La CPU controla todo directamente y debe "esperar" a que el dispositivo termine.
- 2 **E/S mediante Interrupciones:** La CPU ordena la tarea y sigue trabajando. El módulo "interrumpe" a la CPU cuando termina.
- 3 **Acceso Directo a Memoria (DMA):** Un procesador especializado transfiere grandes bloques de datos directo a la memoria sin molestar a la CPU principal.



**Figura 7.1.** Módulo genérico de un módulo E/S.

Figure: Módulo genérico.

# Dispositivos Externos

**Clasificación de Dispositivos Externos (Periféricos)** Se conectan al Módulo de E/S mediante un enlace.

## Interacción con Humanos

Comunicación Usuario  $\leftrightarrow$  Computador (Pantallas, impresoras).

## Interacción con Máquinas

Comunicación Equipo  $\leftrightarrow$  Equipo (Discos magnéticos, sensores).

## De Comunicación

Intercambio con dispositivos remotos (Terminales, redes).

# Estructura de un Dispositivo Externo

- **Los datos:** Se intercambian como bits enviados a, o recibidos desde, el módulo de E/S.
- **Señales de Control:** Determinan la función (ENTRADA, SALIDA, Estado).
- **Buffer:** Almacén temporal (aprox 8-16 bits) para compensar velocidades.

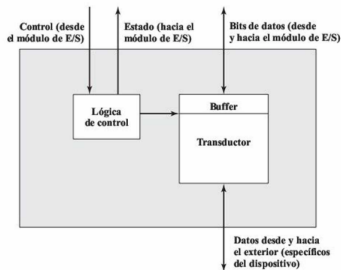


Figura 7.2. Diagrama de bloques de un dispositivo externo.

**Figure:** Figura 7.2: Diagrama de bloques interno de un dispositivo externo. Diagrama de bloques interno de un dispositivo externo.

- El teclado funciona como un dispositivo de entrada, mientras que el monitor actúa como un dispositivo de salida.
- La unidad básica de intercambio de información es el carácter. Cada carácter está asociado a un código, generalmente representado mediante 7 u 8 bits.
- Uno de los códigos más utilizados es el código IRA (International Reference Alphabet), también conocido como ASCII.



Figure: Ejemplo de dispositivo de interacción con humanos (Teclado).

# Código ASCII (IRA)

- El código ASCII (IRA – International Reference Alphabet) es un estándar que asigna un valor numérico a cada carácter.
- Cada carácter se representa mediante un patrón binario de 7 bits, permitiendo un total de 128 caracteres.
- Existen dos tipos de caracteres:
  - **Imprimibles:** letras, números y símbolos.
  - **De control:** utilizados para funciones como salto de línea, tabulación o retorno de carro.
- Este código permite que el teclado, el monitor y el computador intercambien información de forma uniforme.



# Código ASCII (IRA)

bit posición											
b <sub>7</sub>				0	0	0	0	1	1	1	1
b <sub>6</sub>				0	0	1	1	0	0	1	1
b <sub>5</sub>				0	1	0	1	0	1	0	1
b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>								
0	0	0	0	NUL	DLE	SP	0	@	P	'	p
0	0	0	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	STX	DC2	"	2	B	R	b	r
0	0	1	1	ETX	DC3	#	3	C	S	c	s
0	1	0	0	EOT	DC4	\$	4	D	T	d	t
0	1	0	1	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	ACK	SYN	&	6	F	V	f	v
0	1	1	1	BEL	ETB	'	7	G	W	g	w
1	0	0	0	BS	CAN	(	8	H	X	h	x
1	0	0	1	HT	EM	)	9	I	Y	i	y
1	0	1	0	LF	SUB	*	:	J	Z	j	z
1	0	1	1	VT	ESC	+	;	K	[	k	{
1	1	0	0	FF	FS	,	<	L	\	l	
1	1	0	1	CR	GS	-	=	M	]	m	}
1	1	1	0	SO	RS	.	>	N	^	n	~
1	1	1	1	SI	US	/	?	O	_	o	DEL

Figure: Ejemplo de codificación de datos y bits de posición.

# Controlador de Disco (Disk Drive)

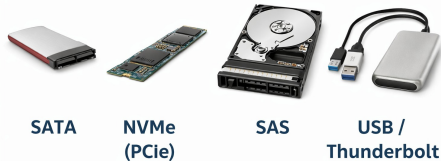
- El controlador de disco es el componente encargado de gestionar la comunicación entre el módulo de E/S y el almacenamiento secundario.
- Se encarga de controlar el mecanismo de lectura/escritura y el movimiento del brazo.
- Utiliza un buffer para almacenar temporalmente los datos.
- El transductor convierte patrones magnéticos en señales eléctricas.



Figure: Ejemplo (Disco Duro).

# Controladores de Disco (Interfaces actuales)

- **SATA (Serial ATA)**: Interfaz ampliamente utilizada para discos duros (HDD) y unidades de estado sólido (SSD). Ofrece una buena relación entre costo, capacidad y velocidad.
- **NVMe (Non-Volatile Memory Express)**: Interfaz moderna que funciona sobre el bus PCI Express. Diseñada específicamente para SSD, permite velocidades de transferencia muy superiores a SATA.
- **SAS (Serial Attached SCSI)**: Utilizada principalmente en servidores y centros de datos. Ofrece alta fiabilidad, mayor velocidad y soporte para múltiples dispositivos.
- **USB / Thunderbolt**: Interfaces empleadas para dispositivos de almacenamiento externo. Permiten conexión rápida y portátil de discos duros y SSD.



- **Definición:** Los módulos de E/S son un tipo de intermediario inteligente entre el procesador y los periféricos. Los módulos permiten ignorar el funcionamiento físico de cada dispositivo externo, limitándose a solo ejecutar órdenes simples, garantizando una comunicación eficiente y transparente entre el sistema y periféricos.

## Control y Temporización (Control & Timing)

- **Objetivo:** Coordinar el flujo de tráfico entre recursos internos y dispositivos externos.
- **Escenario de transferencia (5 pasos):**
  - 1 CPU interroga al módulo para verificar el estado.
  - 2 El módulo devuelve el estado del dispositivo.
  - 3 Si está listo, la CPU envía un comando de transferencia.
  - 4 El módulo obtiene el dato (8 o 16 bits) del periférico.
  - 5 Los datos se transfieren del módulo a la CPU.

## Comunicación con el Procesador Involucra el uso del bus del sistema para cuatro tareas:

- **Decodificación de comandos:** Aceptar señales de control (ej. READ SECTOR, SEEK, WRITE).
- **Intercambio de datos:** Movimiento de bits sobre el bus de datos.
- **Informe de estado:** Señales para reportar disponibilidad (ej. BUSY o READY).
- **Reconocimiento de dirección:** Identificar la dirección única asignada a cada periférico.

## Comunicación con el Dispositivo

- El módulo debe realizar la interfaz directa con el periférico.
- Gestiona el envío de comandos específicos del hardware.
- Recibe información de estado y datos propios del dispositivo.

## Almacenamiento Temporal (Data Buffering)

- **Problema:** La velocidad de la memoria es órdenes de magnitud mayor a la de los periféricos.
- **Solución:** El módulo actúa como un "reservorio".
  - Recibe datos de la memoria en ráfagas rápidas.
  - Los envía al periférico a su ritmo lento (y viceversa).
- **Beneficio:** No se bloquea la memoria durante operaciones lentas.

**Detección de Errores** El módulo reporta fallos a la CPU:

- **Errores físicos:** Mal funcionamiento mecánico o eléctrico (ej. atasco de papel, sector defectuoso).
- **Errores de transmisión:** Cambios accidentales en los patrones de bits.
  - **Ejemplo:** Uso de bits de paridad (par o impar) para verificar la integridad de cada carácter.

En los módulos la complejidad y cantidad de dispositivos externos varían considerablemente según el dispositivo.

- **Conexión con el sistema:** El módulo de E/S se conecta al computador mediante el bus del sistema.
  - Líneas de datos
  - Líneas de dirección
  - Líneas de control
- **Registros:**
  - Registro de datos: Los datos que se transfieren a, y desde, el módulo se almacenan temporalmente en los registros.
  - Registro de estado y control: Contiene la información sobre el estado actual del dispositivo y puede recibir información de control del procesador.



# Estructura de Módulos de E/S

- **Lógica de E/S:**

- El módulo interactúa con el procesador a través de líneas de control. En donde se interpreta las órdenes del procesador. También, pueden ser utilizadas por el modulo para señales de arbitraje y estado.
- Además, el modulo debe reconocer y generar direcciones únicas para los dispositivos.
- Tiene una lógica específica para cada dispositivo a su control.
- El módulo oculta detalles al procesador como: temporización, formato de datos y operaciones electromecánicas.

- **Tipos de modulos de E/S:**

- Canal de E/S o procesador de E/S: Se encarga de la mayoría de detalles, generando una interfaz de alto nivel.
  - Se utiliza en grandes computadores.
- Controlador de E/S o controlador de dispositivo: Es simple y requiere un control detallado normalmente.
  - Se utilizan en microcomputadores.

# Ejemplo de un módulo de E/S

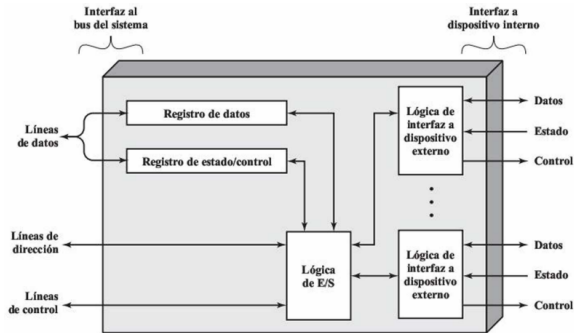


Figura 7.3. Diagrama de bloques de un módulo de E/S.

Figure: Ejemplo de del Intel 82C55A.

- La E/S Programada es la técnica para realizar operaciones de entrada y salida donde los datos se intercambian entre el procesador y el módulo de E/S.
- Esta técnica tiene un control directo sobre la operación de E/S
  - **Ejecución de órdenes:** Cuando el procesador encuentra una instrucción de E/S, envía una orden al módulo correspondiente para que realice la acción.
  - **Ausencia de interrupciones:** El módulo de E/S realiza la tarea y activa bits en su registro de estado, pero no avisa al procesador cuando termina.
  - **Responsabilidad del procesador:** Debido a que no hay avisos automáticos, el procesador debe comprobar periódicamente el estado del módulo hasta determinar que la operación ha finalizado.
  - **Sincronización manual:** Si el procesador es más rápido que el módulo de E/S, debe esperar y desperdiciar tiempo hasta que la E/S concluya.
  - **Gestión de memoria:** El procesador es el encargado de extraer los datos de la memoria (en una salida) o de almacenarlos (en una entrada).

- Para ejecutar una operación, el procesador proporciona una dirección y una de las cuatro órdenes de E/S posibles:
  - **Control:** Se usa para activar el periférico e indicarle qué tarea realizar. Por ejemplo, ordenar a una unidad de cinta que rebobine.
  - **Test:** Sirve para que el procesador verifique el estado del módulo y sus periféricos. Permite saber si el dispositivo está disponible, si la operación anterior terminó o si hubo errores.
  - **Lectura:** El módulo capta un dato del periférico y lo coloca en un buffer interno. Luego, el procesador solicita que el módulo ponga ese dato en el bus de datos para obtenerlo.
  - **Escritura:** El módulo capta un dato desde el bus de datos y lo transmite al periférico correspondiente.

- Las instrucciones de E/S permiten al procesador comunicarse con los dispositivos externos a través de módulos de E/S. En la E/S programada existe una relación directa entre las instrucciones ejecutadas por la CPU y las órdenes enviadas al dispositivo, generalmente de uno a uno.
- Cada dispositivo de E/S posee un identificador o dirección única. Cuando el procesador envía una orden de E/S, incluye esta dirección para que el módulo correspondiente interprete si la operación está dirigida a él.
- Cuando la memoria principal y los dispositivos de E/S comparten un bus común, se utilizan dos esquemas de direccionamiento: E/S asignada en memoria y E/S aislada.

# Direccionamiento de E/S: Memory-Mapped vs E/S Aislada

En la E/S asignada en memoria (memory-mapped), los registros de datos y estado de los dispositivos se consideran posiciones de memoria. Esto permite usar el mismo conjunto de instrucciones máquina para acceder tanto a memoria como a E/S, facilitando la programación.

- En la E/S aislada, la memoria y la E/S utilizan espacios de direcciones separados y los dispositivos solo pueden ser accedidos mediante instrucciones específicas de E/S. Aunque conserva el espacio de direcciones de memoria, ofrece un número más limitado de instrucciones.
- La E/S asignada en memoria ofrece mayor flexibilidad y eficiencia en la programación, mientras que la E/S aislada evita el uso del espacio de memoria. Por ello, ambos métodos se emplean comúnmente según el diseño del sistema.

# Ejemplo gráfico



Figure: Comparación entre E/S asignada en memoria y E/S aislada

# E/S mediante Interrupción

- En la E/S programada, la CPU debe esperar a que el dispositivo esté listo, reduciendo el rendimiento del sistema.
- Con la E/S mediante interrupciones, el procesador envía la orden de E/S y continúa ejecutando otras tareas.
- Cuando el dispositivo está preparado, el módulo de E/S genera una interrupción.
- El procesador atiende la interrupción, realiza la transferencia de datos y luego continúa su ejecución normal.
- Con ese método aprovecha de mejor manera el tiempo del procesador, mejorando el rendimiento del sistema.



# Procesamiento de la Interrupción

Cuando un dispositivo de E/S termina una operación de E/S, se produce la siguiente secuencia de eventos en el hardware:

- **Paso 1:** El dispositivo envía una señal de interrupción al procesador
- **Paso 2:** El procesador finaliza la instrucción en ejecución.
- **Paso 3:** El procesador comprueba la existencia de interrupciones e identifica la interrupción activa. Posteriormente, envía una señal de reconocimiento (**acknowledge**) al dispositivo que originó la interrupción, provocando que este desactive su señal de interrupción.

## Paso 4

El procesador guarda el contexto del programa interrumpido en la pila de control del sistema, incluyendo:

- El estado del procesador se guarda en la Palabra de Estado del Programa (PSW, **Program Status Word**).
- La dirección de la siguiente instrucción a ejecutar, contenida en el Contador de Programa (PC).

## Paso 5

- El procesador carga el Contador de Programa (PC) con la dirección de inicio de la Rutina de Servicio de Interrupción (ISR).
- El control pasa a la Rutina de Servicio de Interrupción.

# Ejecución de la Rutina de Servicio de Interrupción

- **Paso 6:** La ISR guarda los registros del procesador en PC. Este paso asegura la preservación completa del estado del programa interrumpido.
- **Paso 7:** La rutina de servicio de interrupción procesa el evento que originó la interrupción. Esto puede incluir:
  - Examinar el estado del dispositivo de E/S.
  - Transferir datos entre el dispositivo y la memoria.
  - Enviar señales o comandos adicionales al dispositivo para finalizar la operación de entrada/salida.

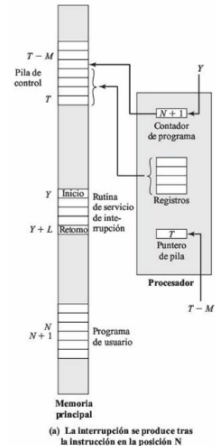
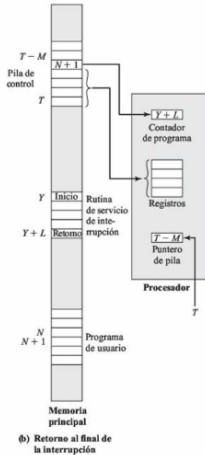


Figure: La interrupción se produce tras la instrucción en la posición N.

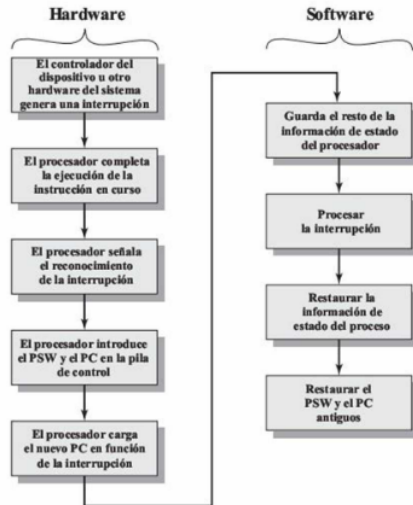
# Restauración del Contexto y Retorno al Programa



- **Paso 8:** Se restauran los registros desde la pila de control y se restauran los registros del procesador
- **Paso 9:** Se recuperan PSW y PC, reanudando la ejecución del programa.

Figure: Retorno al programa tras la atención de la interrupción.

# Diagrama Procedimiento de Interrupción del Sistema



En la E/S mediante interrupciones aparecen dos problemas fundamentales:

- 1 Identificar qué dispositivo ha generado la interrupción.
- 2 Decidir cuál interrupción debe atender primero la CPU cuando existen varias solicitudes.
- 3 Esto ocurre debido a que normalmente existen múltiples módulos de E/S, el procesador necesita mecanismos eficientes para reconocer el origen de cada interrupción.

## Técnicas de identificación del dispositivo

- **Múltiples líneas de interrupción:** La aproximación más directa consiste en proporcionar varias líneas de interrupción entre el procesador y los módulos de E/S. Sin embargo, no es práctico dedicar más de unas pocas líneas del bus o de los terminales del procesador a este fin; cuando se utilizan varias líneas, es probable que a cada una se conecten varios módulos de E/S.
- **Consulta por software (software poll):** Cuando el procesador detecta una interrupción, se produce una bifurcación a una rutina de servicio de interrupción que consulta a cada módulo de E/S para determinar cuál ha provocado la interrupción. Esta identificación puede realizarse mediante una línea específica de prueba o mediante la lectura de registros de estado direccionables.

- **Daisy Chain:**

Daisy Chain es una técnica que funciona como un sondeo por hardware. Cuando el procesador detecta una interrupción, envía una señal de reconocimiento que se va propagando por los módulos hasta llegar al que realizó la solicitud, el cual responde colocando en las líneas de datos un vector que lo identifica de forma única o indica la dirección del módulo. El procesador utiliza este vector para acceder directamente a la rutina de servicio correspondiente al dispositivo, evitando ejecutar primero una rutina general de interrupciones, lo que hace que este mecanismo, conocido como interrupción vectorizada, sea más rápido y eficiente.

- **Arbitraje de Bus:**

Un módulo de E/S debe en primer lugar disponer del control del bus antes de poder activar la línea de petición de interrupción. Así, solo un módulo puede activar la línea en un instante. Cuando el procesador detecta la interrupción, responde mediante la línea de reconocimiento de interrupción. Después, el módulo que solicitó la interrupción sitúa su vector en las líneas de datos.

## El Problema (Inconvenientes E/S)

La E/S programada y con interrupciones requiere intervención activa de la CPU, lo que trae dos problemas:

- La velocidad de transferencia se limita a la velocidad de respuesta del procesador.
- El procesador pierde tiempo ejecutando instrucciones de gestión por cada dato transferido.

## La Solución

Para transferir grandes volúmenes de datos se usa el **DMA**, una técnica que evita que los datos pasen a través del procesador, liberándolo para otras tareas.



# Concepto y Necesidad del DMA

- **El Problema:** La E/S Programada y por Interrupciones limita el rendimiento porque la CPU debe gestionar cada dato ("Busy-waiting" o "Context Switching").
- **La Solución (DMA):** Un módulo especializado que toma el control del bus para transferir bloques completos directamente a memoria.
- **Diferencia:** La CPU solo interviene al inicio (delegación) y al final (interrupción por finalización).

Table 8.1 I/O Techniques

	No Interrupts	Use of Interrupts
I/O-to-memory transfer through processor	Programmed I/O	Interrupt-driven I/O
Direct I/O-to-memory transfer		Direct memory access (DMA)

**Figure:** Tabla 8.1: El DMA es la única técnica con acceso directo a memoria.

# Arquitectura Interna del Módulo

El DMA actúa como un procesador simplificado. Para operar independientemente usa registros clave:

- **Address Register (AR):** Apunta a la dirección de memoria de destino/origen.
- **Data Count (DC):** Cuenta cuántas palabras faltan por transferir.
- **Control Logic:** Gestiona el "Handshake" (DREQ, DACK, HRQ, HLDA).

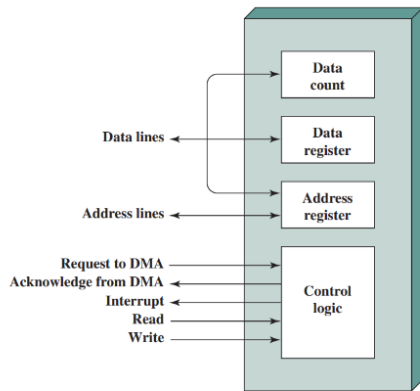


Figure 8.12 Typical DMA Block Diagram

Figure: Figura 8.12: Registros internos y lógica de control.

# Funcionalidad: Robo de Ciclo (Cycle Stealing)

- El DMA **no** ejecuta instrucciones, solo transfiere datos.
- **Cycle Stealing:** El DMA pausa a la CPU durante **un ciclo de bus** para transferir una palabra.
- **Ventaja:** No es una interrupción (no guarda contexto), por lo que la CPU sigue trabajando internamente (caché/registros) casi sin impacto.

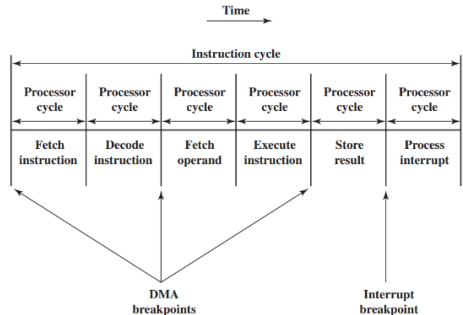


Figure 8.13 DMA and Interrupt Breakpoints during an Instruction Cycle

Figure: Figura 8.13: El DMA se inserta dentro del ciclo de instrucción.

# Configuraciones de Interconexión

La ubicación del DMA define la eficiencia:

- 1 **Bus Único:** Todos comparten el bus. Ineficiente (doble ciclo por dato).
- 2 **Bus de E/S (Recomendado):** El DMA aísla el tráfico de los periféricos en un bus privado, liberando el bus del sistema para la CPU y la Memoria.

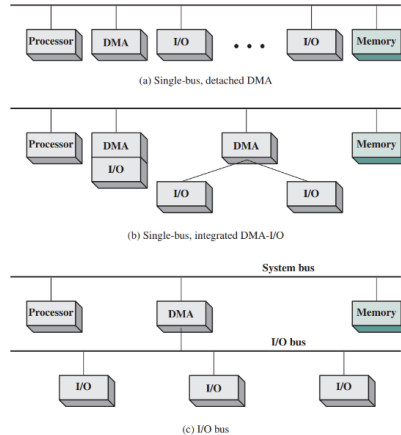


Figure 8.14 Alternative DMA Configurations

Figure: Figura 8.14: Evolución hacia el Bus de E/S (c).

# Funcionalidad de DMA

## Delegación de la Orden

El procesador envía una orden al módulo DMA con la información clave: si es lectura/escritura, la dirección del dispositivo, la posición en memoria y la cantidad de datos.

## Robo de Ciclo (Cycle Stealing)

El DMA toma el control del bus imitando al procesador. Fuerza a la CPU a suspenderse temporalmente (un ciclo) para transferir datos directamente a la memoria.

## Eficiencia

El procesador continúa trabajando en paralelo y solo recibe una interrupción al final, cuando el bloque entero ya se transfirió.

# ¡Muchas Gracias!

¿Preguntas?

ricardo.leon02@epn.edu.ec, andre.rosero@epn.edu.ec, mauro.valencia@epn.edu.ec; david.paredes01@epn.edu.ec;  
luzsamara.salazar@epn.edu.ec