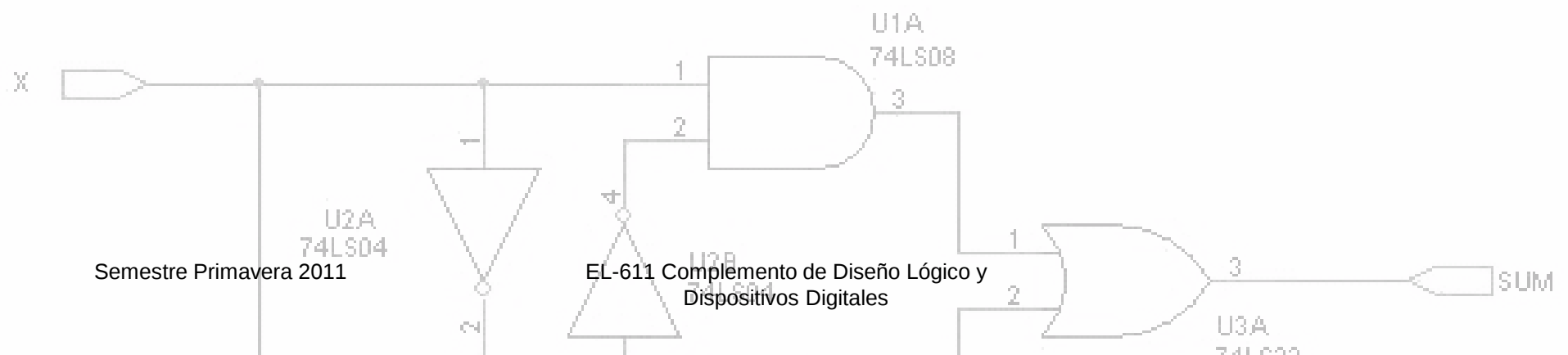




Objetivos y Evaluación

- Segundo Curso de Sistemas Digitales
 - Complementar Materia
 - Enfoque Diseños de Mayor Envergadura
- 1 Control + Examen y 6 Ejercicios (aprox.)
- Tareas (Diseños)
- Trabajo Individual Disertación Personal
 - Presentación
 - Asistencia
 - Informe

Un poco de historia ...

- 1850: George Boole inventa el Algebra Booleana
 - relaciona proposiciones lógicas con símbolos
 - permite la manipulación de expresiones lógicas utilizando matemáticas
 - 1938: Claude Shannon vincula el Algebra de Boole con los conmutadores (“switches”)ul> - su Tesis de Master
- 1945: John von Neumann desarrolla el primer computador con almacenamiento
 - sus elementos de “switching” eran tubos (“vacuum tubes”; un gran avance a los relays)
- 1946: ENIAC . . . El primer computador completamente electrónico
 - 18,000 tubos
 - varios cientos de multiplicaciones por minuto
- 1947: Shockley, Brittain, and Bardeen inventan el transistor
 - reemplaza a los tubos
 - permite la integración de múltiples dispositivos en un paquete
 - puerta a la electrónica moderna

¿Qué es el diseño lógico?

➤ ¿Qué es diseño?

- dada una especificación de un problema, obtener la forma de solucionarlo eligiendo las componentes apropiadas de un conjunto de componentes disponible
- cumpliendo, al mismo tiempo, algunos criterios de tamaño, costo, consumo de energía, belleza, elegancia, etc.

➤ ¿Qué es diseño lógico?

- determinar el conjunto de componentes lógicas digitales para realizar un control y/o manipulación de datos y/o funciones de comunicación especificadas y la interconexión entre ellas
- cuales componentes lógicas seleccionar? – hay muchas tecnologías de implementación (por ejemplo, componentes de funciones fijas estándares, dispositivos programables, transistores en un chip, etc.)
- el diseño puede necesitar ser optimizado y/o transformado para cumplir con las restricciones del diseño

¿Que pasa con el diseño digital en la actualidad?

- Importantes tendencias de como la industria realiza diseño de hardware
 - diseños cada vez mas grandes
 - tiempo de introducción al mercado cada vez mas corto
 - productos cada vez mas baratos
- Escala
 - amplio uso herramienta de diseño con ayuda del computador por sobre diseños manuales
 - niveles múltiples de representación de diseños
- Tiempo
 - énfasis en representaciones abstractas de diseño
 - componentes de funciones programables por sobre componentes de funciones fijas
 - técnicas de síntesis automáticas
- Costos
 - niveles cada vez mas alto de integración
 - uso de simulación para la depuración de los diseños
 - simular y verificar antes de fabricar

Sistemas Digitales

- Sistemas Analógicos / Digitales
- Sistemas Digitales Binarios
- Predominio de los Sistemas Digitales



Ventajas de los Sistemas Digitales

- Menos propenso a **errores** dado que las variables sólo tienen dos valores
- La **exactitud** depende del número de bits utilizados en la representación numérica
- La representación digital permite el procesamiento de **información numérica** como **no-numérica**
- Procesos se descomponen fácilmente en **sub-tareas** ejecutables en secuencias
- Sistema de **propósitos generales** (computador)
- Desarrollo de sistemas complejos a **bajo costo**

Sistemas Digitales (Binarios o “Multiple-valued”)

Máquinas Combinacionales

Máquinas Secuenciales

Lógica “Random”

Lógica de Arreglo

Lógica Programada
Microprocesadores
Controladores
Microprogramados

Síncrona

Asíncrona

SSI

MSI

LSI/
VLSI

ROM

PLA

Modo Nivel

Modo Pulso

Contemporáneo

Tradicional

Cuasi-Síncrono

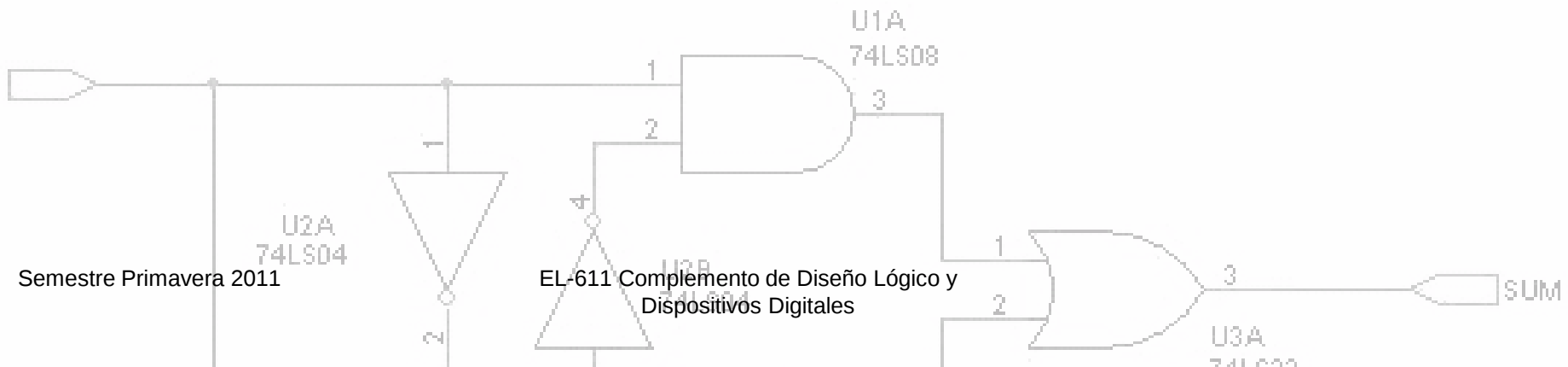
No requieren Memoria

Requieren Memoria

Diseño

➤ Enfoque Tradicional

- Diagramas de Estados
- Tabla de Transición de Estados
- Minimización de Estados
- Asignación de Estados
- Variables de Excitación y de Estado
- Implementación “Biestable-Compuerta”



Diseño

➤ Enfoque Contemporáneo

➤ División sistémica (algorítmico)

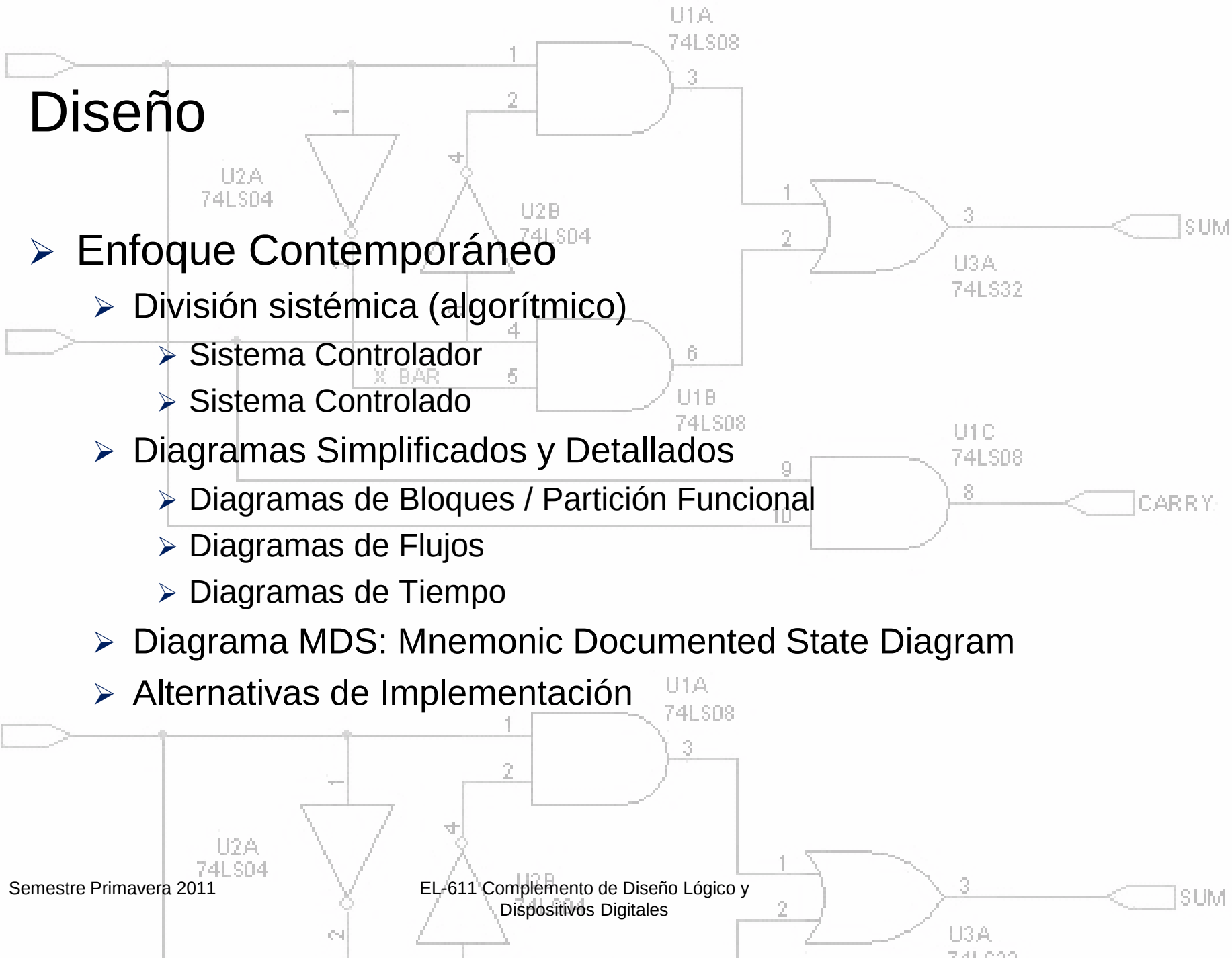
- Sistema Controlador
- Sistema Controlado

➤ Diagramas Simplificados y Detallados

- Diagramas de Bloques / Partición Funcional
- Diagramas de Flujos
- Diagramas de Tiempo

➤ Diagrama MDS: Mnemonic Documented State Diagram

➤ Alternativas de Implementación



Modelos de Diseño

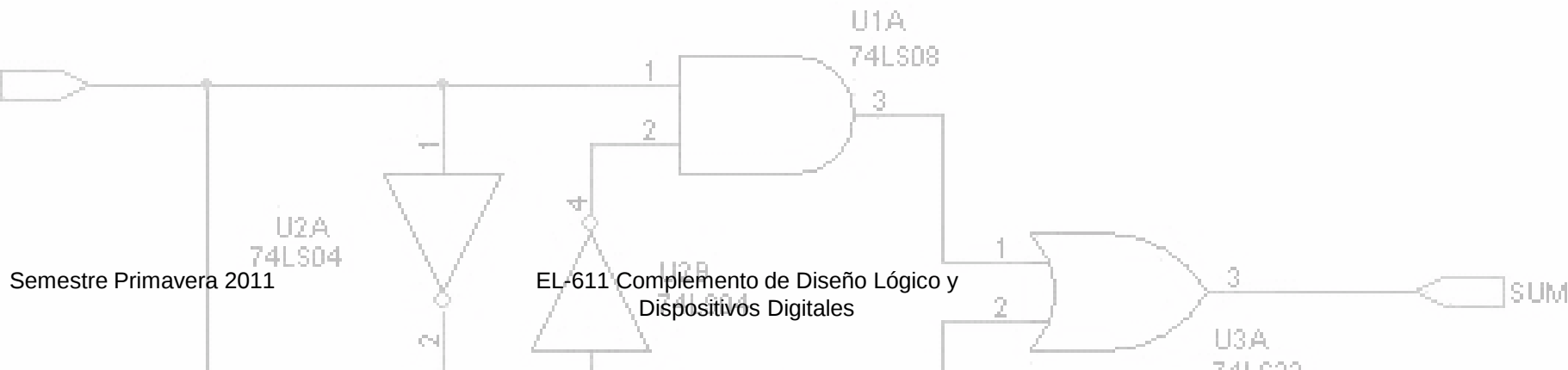
- Modelo **Sistémico**: caja negra
- Modelo **Algorítmico**: el sistema digital se ve efectuando un cálculo, una “computación”, una transformación de datos (“data objects”) o vectores de variables binarias (“bit-vectors”)
 - Si el cálculo es complejo se descompone en cálculos más simples descritos por un **algoritmo**
 - Consiste de los siguientes elementos:
 - **Almacenamiento** (“storage”) para la representación de los vectores
 - **Operadores** para realizar los cálculos primitivos en los datos
 - **Control** de la secuencia de los cálculos primitivos

Modelo Algorítmico

- Los algoritmos ejecutados por el sistema pueden ser **fijos o programables** (cambiables)
- El diseño consiste en especificar:
 - Descripción estructural: la **estructura** e interconexión de las componentes
 - Descripción del comportamiento: **algoritmos** específicos a ser ejecutados por el sistema
 - **Formatos** de representación de los datos y algoritmos

Ejemplo Diseño con un enfoque algorítmico

- Considerar el cálculo para obtener el máximo entre n enteros. Cada entero es representado como un vector de variables binarias (“bit-vectors”), y los vectores son almacenados en componentes de almacenamiento. El resultado se guarda en un registro *MAX*.
- Los operadores primitivos son los siguientes



U1A
741 002

U1
741 002

READFIRST obtains the first number from storage and stores it in register *MAX*
READNEXT obtains the next number from storage and stores it in register *TEMP*
COMPARE compares *TEMP* and *MAX* and stores 1 in *COMP* if *TEMP* > *MAX* and 0 if *TEMP* ≤ *MAX*
EXCHANGE exchanges *TEMP* and *MAX*

In addition to these data operations, the following control operations are used:

SETCOUNT initializes *COUNT* to 1
COUNTUP increments *COUNT* by 1

3A
741 002

1C
741 002

SUM

CARRY

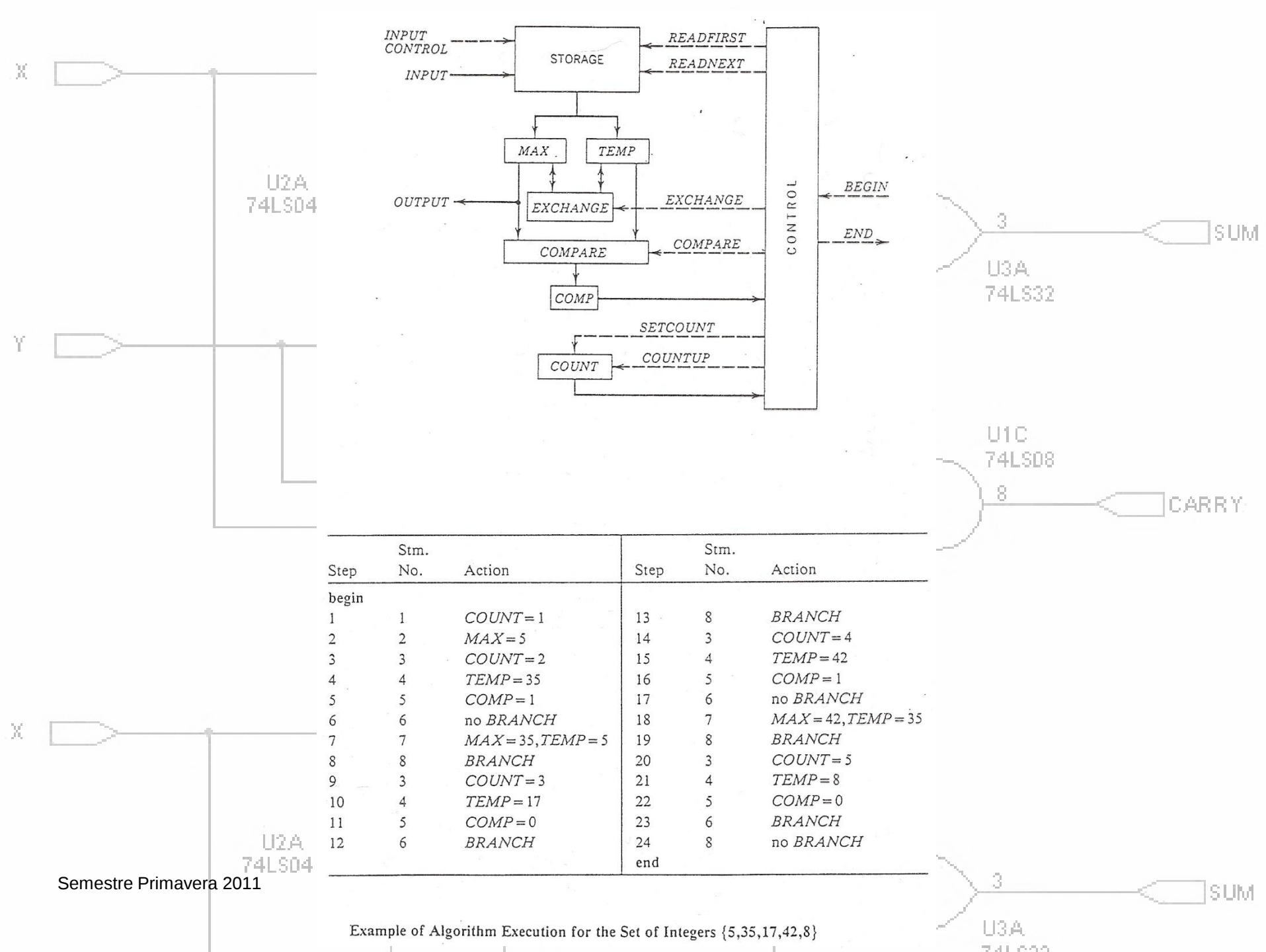
```
begin
  1: SETCOUNT
  2: READFIRST
  3: COUNTUP
  4: READNEXT
  5: COMPARE
  6: BRANCH if COMP = 0 to 8
  7: EXCHANGE
  8: BRANCH if COUNT ≠ n to 3
end
```

EL-611 Complemento de Diseño Logico y
Dispositivos Digitales

U2
741 002

U3A
741 002

SUM



Tecnologías Sistemas Combinacionales

➤ Componentes Estándares

- SSI (10 compuertas)
- MSI (10 – 100 compuertas)

➤ ROM's y PLA's/PAL's: Programmable Logic Array

➤ ASIC's: Application Specific Integrated Circuits (100.000 compuertas)

➤ FPGA: Field Programmable Gate Arrays

TABLE 4.1

Technology Advancements of the Intel Microprocessor Family, 1971–2003

Microprocessor	Year	Application	Number of Transistors	MIPS (Millions of Instr. Per Sec.)	Clock Speed Range
4004	1971	Calculator	2,300	0.1	108 KHz
8008	1972	Character terminal	3,500	0.1	200 KHz
8080	1974	Altair 8080 Homebrew PC	6,000	0.6	2 MHz
8086 (8088)	1978	Business PC (IBM PC)	29,000	0.8	5–10 MHz
80286	1982		134,000	3	6–12.5 MHz
80386	1985		275,000	10	16–33 MHz
80486	1989	Home PC	1.2 million	40	25–50 MHz
Pentium	1993		3.1 million	100	60–66 MHz
Pentium Pro	1995		5.5 million	200	150–200 MHz
Pentium II	1997		7.5 million	300	200–300 MHz
Pentium III	1999	High-performance workstations	9.5 million	500	333–900 MHz
Pentium IV	2001	Web servers, Graphics apps	42.0 million	1500	1–1.5 GHz
Pentium IV	2003	Multi-media PC, Hyperthreading	55.0 million	3000	2–3.5 GHz

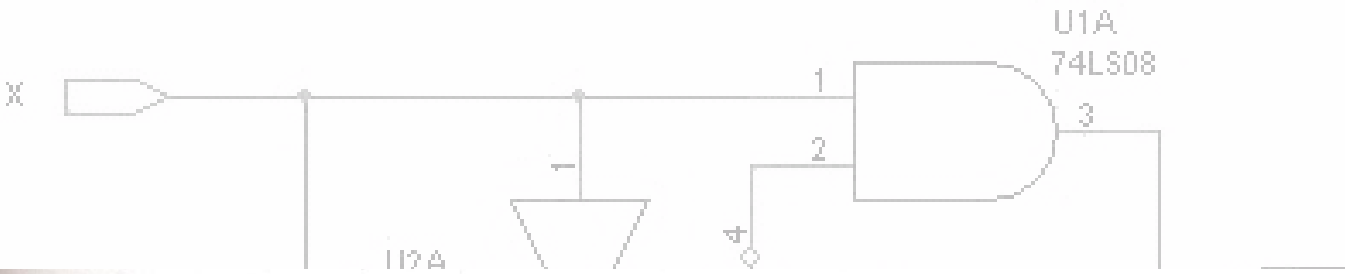


TABLE 4.2

Historical Development of Components (Adapted from Oldfield and Dorf, *Field Programmable Gate Arrays*, John Wiley, New York, 1995)

	1960s SSI/MSI	1970s LSI	1980s VLSI	1990s Programmable Logic
Components	Logic, Resistor/ Transistor elements	8-bit μ processor, Memory, ROM	32-bit μ processor, Gate arrays	64-bit μ processor, PALs, FPGAs
Complexity Level (# of gates)	100s	10,000s	1 million	100,000s to millions
Pervasive Components	TTL 7400 series	Intel 8008, ROM	Intel 8086, Motorola 68000, Gate arrays, PALs	Pentium I, II, III, FPGAs, Complex PLDs
Dominant Trend	Standard catalog of components	Larger, General-purpose components, e.g., Micro- processors and Random- access memories	Application-spe- cific integrated circuits	Field- programmable components

