

Departamento de Arquitectura      Instituto de Computación  
Universidad de la República  
Montevideo - Uruguay

# **Notas de Teórico**

## **Máquina Lógica General**

**Arquitectura de Computadoras**  
(Versión 4.3 - 2012)

## 12 MAQUINA LOGICA GENERAL

### 12.1 Introducción

En este capítulo veremos el esquema general de una máquina lógica capaz de implementar cualquier máquina lógica particular e introduciremos el concepto de computadora, para en los capítulos siguientes concentrarnos en la forma de construirlas (su arquitectura).

En los cursos de Teoría de Computación se analiza la teoría general de los autómatas y, en particular, se analiza el hecho que todo problema resoluble mediante un algoritmo (o sea "computable") se puede reducir al problema de traducir símbolos y/o reconocer secuencias. En nuestra materia tomaremos esos elementos como un dato y veremos la aplicabilidad de esos conceptos a las máquinas lógicas y sus implicancias.

### 12.2 Máquina Lógica General

En el capítulo de Máquinas de Estado, ya vimos que los circuitos secuenciales implementados a partir de las máquinas de estado, en particular las que obedecen al esquema de Mealy, tienen un comportamiento que puede expresarse matemáticamente mediante las expresiones:

$$Y = G(X, E)$$
$$E' = H(X, E)$$

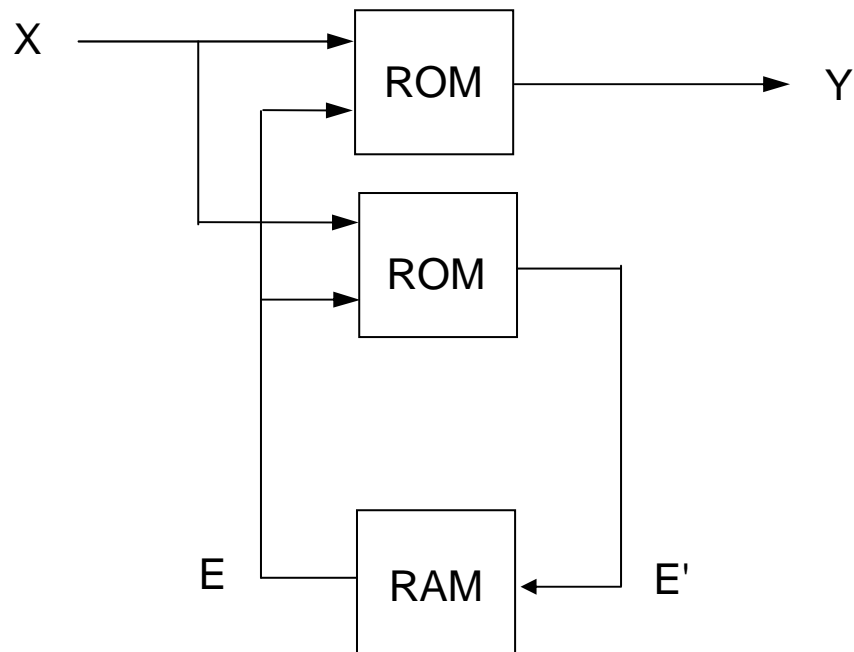
siendo:

Y = salida actual  
E' = próximo estado  
X = entrada actual  
E = estado actual  
G = función de salida  
H = función de estado

La primera observación a realizar es que una máquina combinatoria es un caso particular de máquina secuencial (dónde la función de estado es la identidad y la función de salida solo depende de la entrada X). Por tanto una máquina secuencial se puede comportar tanto como un reconocedor de secuencias (en definitiva de esa característica fue que partimos para modelizarlas como máquinas de estado), como un traductor de símbolos (una máquina combinatoria me devuelve un "símbolo de salida" como resultado de una transformación de un "símbolo de entrada").

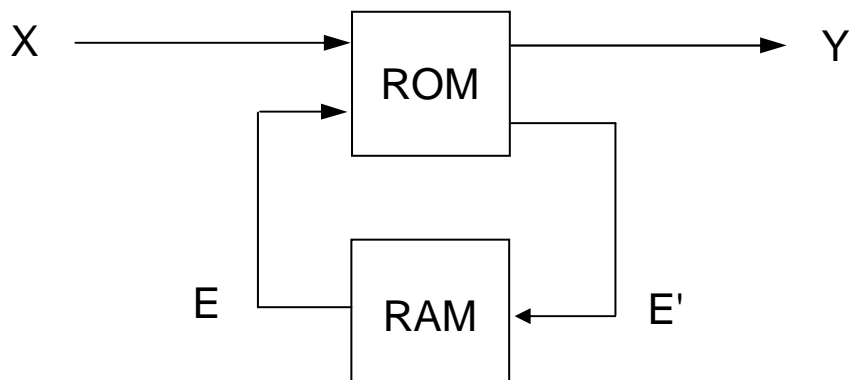
Por tanto una máquina de estados es una máquina lógica general, en el sentido que puede actuar reconociendo secuencias como traduciendo símbolos. De acuerdo a lo expresado en la introducción de este capítulo esto significa que una máquina de estas características es capaz de resolver cualquier problema computable que pueda presentarse.

La segunda observación es que una máquina como ésta puede implementarse utilizando el siguiente esquema de bloques:



donde la ROM de más arriba implementa la función  $G$ , la ROM de abajo la función  $H$  y la RAM es del tipo "registro" para almacenar el estado.

Es fácil ver que el diagrama anterior es equivalente a:



donde hemos juntado las dos ROMs que implementan las funciones  $G$  y  $H$  en una sola.

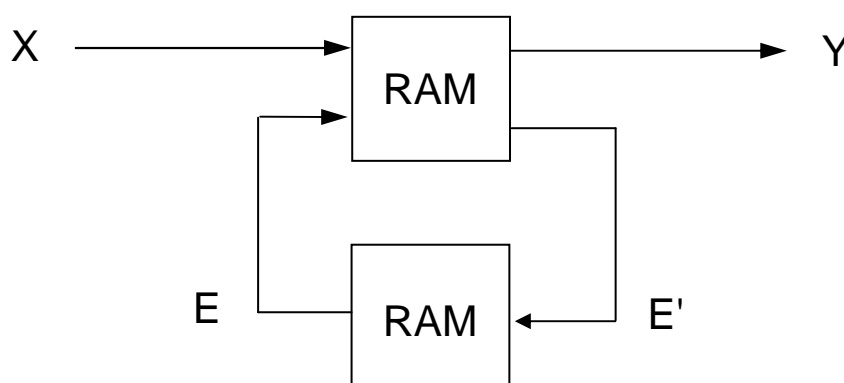
El esquema anterior, que representa una máquina de estados, recibe la denominación de **Máquina Lógica General**, ya que con una ROM de tamaño adecuado (que permita implementar las funciones  $G$  y  $H$  requeridas) y una RAM ("latch" ó registro) con la cantidad de bits adecuada, se puede construir la máquina que resuelva cualquier problema computable específico que se plantee. En otras palabras: dado un problema existe una ROM y una RAM que permiten implementar la máquina que resuelve dicho problema, con la única condición que el mismo sea resoluble por un autómata finito. La condición de "resoluble por un autómata" está dada porque una máquina así resuelve problemas reducibles al reconocimiento de secuencias y/o la traducción de símbolos. La condición de "finito" está vinculada al hecho que la ROM y la RAM pueden tener, al menos desde el punto de vista teórico, el número de bits que se desee, siempre que éste sea finito.

La última observación es que, aunque en principio sorprenda, todo problema "algoritmizable", desde multiplicar números ó calcular raíces cúbicas, hasta la liquidación de sueldos y el diseño asistido por computadora (CAD), puede ser reducido a su mínima expresión abstracta como una combinación de reconocimiento de secuencias y/o traducción de símbolos.

Por tanto la anterior aseveración sobre la máquina lógica general se puede traducir en que dado un problema cualquiera "resoluble" (multiplicar números o liquidar sueldos, por decir dos ejemplos) puedo construir una máquina que lo haga.

### 12.3 Máquina Lógica General Programable

Si damos un paso más y sustituimos la ROM que implementa las funciones de salida y estado por RAM, nos queda el esquema:



Podemos decir que si construyo una máquina de estas características, la misma podrá resolver todos los problemas tales que sus funciones de salida y estado puedan implementarse con la cantidad de bits disponibles en la RAM superior y los bits de almacenamiento de la RAM inferior alcancen para el número de estados requeridos para esos problemas. Para resolver un problema específico alcanzará con "programar" las funciones G y H en la RAM, almacenando los contenidos apropiados en la misma.

### 12.4 Computadora

Una **computadora**, tal como la conocemos hoy, en definitiva es una **máquina lógica universal programable**. Por tanto el esquema anterior es el de mayor nivel de abstracción y generalidad que podemos proponer para representar una computadora (ó computador, ú ordenador, aquí no vamos a entrar a dirimir ni el posible sexo de estos sistemas ni la vieja disputa entre España y América Latina sobre cuál es su nombre correcto, para este último tema existe un interesante artículo accesible en <http://www.elcastellano.org/ordenado.html>).

Notemos que en este contexto "programar" significa determinar los contenidos de la RAM que implementa las funciones de salida y estado.

Para ser más precisos en el esquema presentado, deberíamos separar la RAM superior en una RAM (grande) más una ROM (pequeña). Esto porque todo computador necesita tener implementado al menos un "programa" básico en ROM, que permita ingresar las funciones G y H (específicas para el problema concreto a resolver) en la RAM, porque recordemos que el contenido de las RAMs se borra al quitarle la energía. Pero en definitiva

es un detalle que no es trascendente en este momento.

## 12.5 Arquitectura de una Computadora

El esquema anterior si bien es, como dijimos, el más general que podemos utilizar para representar una computadora, dista bastante de lo que conocemos cotidianamente por estos equipos. No es sencillo ni directo reconocer en él al "Pentium", el "Core Duo" (o al "procesador" que sea), ni el disco duro; apenas la memoria parece estar asociada de alguna manera a la RAM. Ni hablemos del teclado, el mouse y el monitor de video.....

La razón última de esta aparente inconsistencia está en que el anterior esquema, si bien abstracto y general, es normalmente impracticable, en el sentido que rápidamente se llega a tamaños de RAM requeridos para las funciones G y H y los estados que no son posibles de conseguir con circuitos de memoria reales. Entonces si bien **teóricamente** el esquema es correcto, en la **práctica** no es aplicable.

Para ejemplificar este concepto pensemos simplemente el problema de sumar números muy grandes (supongamos representables en 32 bits), sin acarreo. Es un simple problema de traducción de símbolos (traduzco el "símbolo" formado por la yuxtaposición de la pareja de operadores por el "símbolo" que es su suma de 32 bits). Para la máquina que implemente esta solución se requiere una RAM de 64 bits de entrada y 32 bits de salida. Una RAM así tendría una capacidad de  $2^{64} \times 2^2$  bytes =  $2^6 \times 2^{60}$  bytes = 64 Exabytes (1 Exabyte es un Gigabyte de Gigabytes) o sea algo así como el equivalente de 1000 millones de discos duros de un PC moderno.

Es así que surge el concepto de la arquitectura de las computadoras, que consiste en un conjunto de técnicas que permiten construir máquinas lógicas generales programables en forma práctica. En materia terminológica se distingue la "arquitectura" de un computador de la "organización" de un computador. Se puede decir que la **arquitectura** es la visión funcional (el conjunto de recursos que "ve" el programador), mientras que la **organización** es la forma en que se construye una cierta arquitectura en base a circuitos lógicos.