



Electrónica y programación, una relación que no salta a la vista

Ángel Palacios

Recibido el 26 de febrero de 2009. Aprobado el 11 de mayo de 2009

Resumen

La evolución de la electrónica y la informática ha mostrado que la relación entre estas dos ciencias esta muy ligada. Esta relación no solo es vigente desde la concepción de la arquitectura interna de un microprocesador y su repertorio de instrucciones de máquina provistas para que un desarrollador de software construya sus aplicaciones, sino que ha llegado a niveles superiores como es el caso de los circuitos dispositivos PIC(Circuito Integrado Programable), FPGA (Field Programmable Gate Array), GAL (Generic Array Logic), DSP (Procesador Digital de Señales), PLC (Controlador Lógico Programable), PID (Control Proporcional Integral y Derivativo), entre otros.

Palabras clave

Procesar, instrucción, microprocesador, alu, procesador, lenguaje de máquina, programación, algoritmia.

Abstract

The evolution of electronics and computers to show that the relationship between these two sciences is closely linked. This relationship not only from the current design of the internal architecture of a microprocessor and its repertoire of machine instructions provided for a developer to build software applications, but has reached higher levels such as the PIC circuit devices , FPGA, GAL, DSP, PLC, PID and many others to name but a few.

Key words

Processing, instruction, microprocessor, alu, processor, machine language, programming, algorithmics.

1. Introducción.

En muchos instantes de la formación técnica en electrónica se cuestiona, indaga, discierne y en muchos casos se lucha por comprender que papel juega la programación y la algoritmia en la electrónica moderna. Muchas opiniones y corrientes giran en torno de este sentido; no obstante y aunque no resulta tan obvio como parece, son la electrónica y la programación dos tipos de conocimiento que se relacionan entre sí.

II. Desarrollo del tema.

Para contextualizar las dimensiones de esta relación, es necesario iniciar por una pequeña y resumida evolución de la electrónica, la cual es un saber que inicia como una integración de la física y de los conocimientos relacionados con la electricidad y la electro-tecnia; estos, en un principio fueron objeto de físicos e ingenieros civiles, quienes al profundizar en sus quehaceres encontraron un punto importante de estudio de los procesos de transformación de la energía, específicamente, la energía cinética en eléctrica. Con esa necesidad adquiere sentido, la especificidad del ingeniero electricista, quien se preocupa por todos los detalles y procesos de transformación, generación, transporte y distribución de la electricidad y por ende de la optimización y consumo de la potencia eléctrica. Estos últimos son quienes en compañía de los físicos, lograron mezclar todos aquellos estudios sobre la materia (como es el caso del átomo y su modelo orbital de Bohr) y su composición con el fin de obtener un control de aquellos agentes eléctricos que la componen y específicamente los electrones, de ahí su nombre de "Electrón-ica", por lo tanto y según muchos puntos de vista fueron estos instantes los albores de esta ciencia.

Dentro de la electrónica y como fruto de la mezcla y experimentación de estos saberes surgieron muchos desarrollos que dieron origen a los primeros dispositivos electrónicos como lo son el diodo y el transistor.

Estando en la época del diodo, el transistor y el tubo de vacío, se dio origen a las primeras máquinas que mediante tarjetas perforadas realizaron cómputos y "procesaron instrucciones", valga la pena enfatizar la frase "procesaron instrucciones", ya que ahí sé esta hablando de dos grandes términos que unen la electrónica y la programación como lo son "procesar" e "instrucción".

La electrónica vio sus primeras luces y alcances en este tema cuando los matemáticos vieron la posibilidad de describir el mundo en términos de verdaderos y falsos, y aunque suene bastante extraño y sin entrar en una discusión religiosa, filosófica o moral, el ser humano siempre ha sido un ser que gira en torno al número dos (2) como ocurre con: el hombre y la mujer, el día y la noche, el bien y el mal, el sol y la luna, el Ying y el Yang, el cielo y el infierno, la vida y la muerte, y muchos otros. Esta lista de ejemplos de dualidad puede ser interminable, pero esta dualidad del "verdadero" y el "falso" dio origen a todo lo que en la electrónica conocemos como "electrónica digital", la cual es una derivación y aplicación de la

electrónica referente a la manipulación de dígitos. Estos dígitos son específicamente dos entes (en muchos casos voltajes o corrientes) que se denominaron el cero (0) y el uno (1) lógicos, los cuales son una abstracción del "verdadero" y el "falso" de la matemática y su lógica, pero no por aparente que sea, no siempre verdadero equivale a uno y falso a cero, ya que la electrónica digital permite expresar la ocurrencia de un evento mediante un contundente cero e igualmente la ocurrencia puede darse con un contundente uno. Determinadas las formas de describir el mundo, la electrónica empezó por describir los modelos mas básicos que son los que describen la electrónica digital combinacional (multiplexores, decodificadores, comparadores, etc.), posteriormente surge la electrónica digital secuencial (flip-flops, contadores, registros, máquinas de estado, etc.) como una respuesta a una electrónica digital mas pensante y analítica, ya que evalúa sus estados (entradas y salidas) y al momento de cambiar el valor lógico de su salida tome decisiones para hacerlo.

Con esta última evolución de la electrónica digital surge la electrónica micro procesada, la cual es una integración de la electrónica combinacional y la secuencial mediante un modelo de máquina que realice operaciones como es el caso de la ALU (Arithmetic Logic Unit o Unidad de Operaciones Lógicas y Aritméticas). Es la ALU integrada a otros sistemas combinacionales y secuenciales la que posibilita la creación de sistemas electrónicos digitales que manipulándolos apropiadamente realizan una primer emulación de una máquina que recibe "instrucciones" y las ejecuta o "procesa". La pregunta es: ¿y cómo lo hace?, para dar respuesta a este interrogante, es necesario analizar el modelo de la figura 1, el cual fue la base para la creación de los complejos y muy completos sistemas micro-procesados de hoy en día.

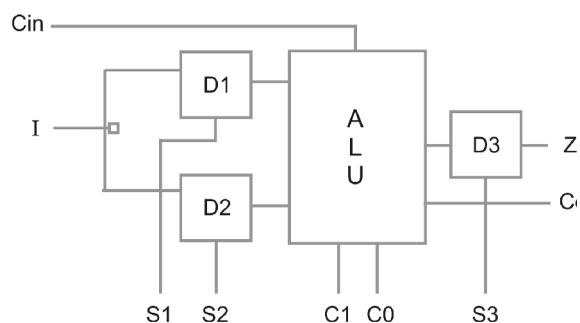


Figura 1. Núcleo de un Microprocesador Fuente: El Autor

Para comprender esta aparentemente pequeña e inofensiva máquina, es necesario contextualizar su alcance; lo primero que se debe indicar es cuales operaciones realiza la ALU. Para este caso se utilizará una ALU que realiza SUMA, RESTA, AND y OR, y aunque suene extraño se verá como se realiza la operación 1

AND 0. Lo primero que se debe aclarar es que tanto el 1 como el 0 se ingresan por el terminal I. Para empezar se debe colocar el 1, valor el cual se ingresa y se guarda en el flip-flop D1 mediante la manipulación del bit seg1, posteriormente se coloca el 0 en I, en ese momento mediante el bit seg2 se ingresa y se guarda en el Flip-Flop D2 el segundo operando de la AND. En este momento ya se puede realizar la AND, solo basta con ajustar los bits C1 y C0 con el código apropiado y en el bit R aparecerá el resultado respectivo. Por último solo resta por almacenar el resultado en el flip-flop D3 mediante el bit Seg3.

Como se acaba de describir, el correcto orden en la manipulación de los bits que controlan esta pequeña máquina, permiten la ejecución y procesamiento (ya se puede comprender la parte de Procesar) de una AND, solo resta por contextualizar la parte de cómo darle ordenes a esta máquina (instrucciones). Para cumplir este cometido, se introduce al modelo una máquina que controle los bits seg1, seg2, C1, C0 y seg3; esta se conoce habitualmente como un decodificador o interprete. Este decodificador es un sistema el cual convertirá un código binario de entrada conocido a los respectivos y apropiados bits que realizan una tarea específica en la ALU y los FF-D (FF o Flip-Flops), y esa tarea es realizar una "Micro-Instrucción" la cual será parte de lo que posteriormente se explicará como "instrucción". Es el momento apropiado de introducir este termino y denotar su alcance no solo a nivel de la electrónica sino también a nivel de la programación. Para explicar como se realiza esto, se debe observar la tabla 1.

La tabla 1 muestra un conjunto de ceros y unos que enmascaran las micro-instrucciones, y este enmascaramiento se realiza de manera tal que cada fila comprende una etapa en la manipulación de los bits que ingresan al interior de la ALU con el fin de que al juntar varios pasos de estos en un orden específico se pueda concretar la denominada "instrucción". Para comprender mejor esto, se hará nuevamente la operación 1 AND 0 mediante las respectivas filas de la tabla con sus respectivos alias o comúnmente llamados mnemónicos:

- FILA 3: Load en D1.
- FILA 7: Storage en D1.
- FILA 0: Load en D2, AND y Load en D3
- FILA 4: Storage en D2 y Storage en D3.

Como es de observar, cada una de las filas seleccionadas realiza la apropiada manipulación de los bits que controlan el sistema ALU y los FF-D, eso sumado al riguroso orden de ejecución de las filas, se puede realizar la operación solicitada. Así solo resta pregun-

tarse: ¿Dónde esta la Instrucción?. Esta pregunta tiene respuesta si se observa que para un programador resulta más provechoso pensar que cuenta con la forma directa de decir AND(1,0), pero esa orden o "instrucción" es transformada mediante un compilador en lenguaje de máquina. Este lenguaje de máquina esta expresado en las micro-instrucciones que al ser ejecutadas en un orden apropiado hacen la solicitada AND.

FILA	I ₁	I ₂	I ₃	MICRO INSTRUCCIONES	S ₁	S ₂	C ₁	C ₂	S ₃
0	0	0	0	AND (D1,D2)	0	1	0	0	1
1	0	0	1	OR (D1,D2)	0	1	0	1	1
2	0	1	0	NOT (D1)	0	0	1	0	1
3	0	1	1	LOAD (D1)	1	0	1	1	0
4	1	0	0	STORAGE - AND	0	0	0	0	0
5	1	0	1	STORAGE - OR	0	0	0	1	0
6	1	1	0	STORAGE - NOT	0	0	1	0	0
7	1	1	1	STORAGE A	0	0	1	1	0

Tabla 1. Equivalencias entre código de Instrucciones, Instrucciones y Bits de Control. Fuente: El autor

La figura 2 muestra un esquema mas generalizado del sistema en el cual ya se cuenta con una ALU de ocho operaciones y por ende su código de instrucciones crece a 16. Este sistema mezcla otros dispositivos y modelos electrónicos que facilitan su funcionamiento y su integración como es el caso de un registro y un decodificador. Estos dispositivos no representan mayor complejidad electrónica pero en cambio son los encargados de respectivamente, recibir el código binario de la instrucción y de activar la fila de la Micro-ROM correspondiente a la instrucción ingresada y activar los bits de control apropiados en el núcleo.

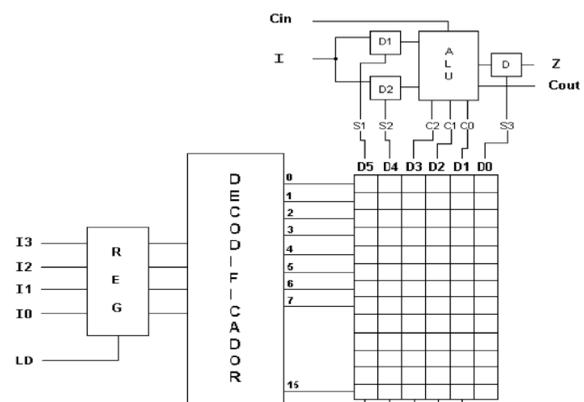


Figura 2. Máquina que Procesa Instrucciones y Operandos. Fuente: El autor

Puede que para un programador una AND no resulte atractiva, pero en cambio un famoso If (num = 3) then tal vez sí, y aunque suene increíble, la máquina mostrada lo puede realizar, no es de olvidar que si se

análogos, conversores y cualquier otra cantidad de recursos según la escala del mismo. Para entender mejor el alcance de este dispositivo, se puede citar el caso de un sistema modem (modulador-demodulador). Es importante recordar que este sistema está presente en todas las comunicaciones y para mayor precisión, se deberá pensar en el modem que tienen equipados los PC's caseros y que hasta hace poco se usó para conectarse a Internet. Este dispositivo tenía la enorme labor de: recibir la señal, filtrarla, amplificarla, reconstruirla, cambiarla de frecuencias a valores digitales, descomprimirla, decodificarla y por último extraer el paquete con los datos a ser recibidos. Esta operación también se realiza en forma contraria cuando el PC trasmite datos a la Internet, por lo tanto, todo el proceso matemático de corrección de error, reconstrucción de la señal, descompresión y decodificación, son procesos que demandan una alta velocidad de procesamiento y operaciones numéricas complejas.

Este es solo un caso de uso de estos dispositivos; no obstante ellos están presentes en sistemas de audio, video y en general, en cualquier sistema que requiera la integración de señales y una velocidad de procesamiento bastante alto.

Un ejemplo muy cotidiano de los DSP y los microcontroladores es el caso de los teléfonos celulares. Este es en esencia un teléfono inalámbrico, y por ende, debe realizar la labor de recibir señales (radio, televisión, microondas, etc.), filtrar la señal de su estación base, amplificarla, reconstruirla y transformar estas ondas electromagnéticas de frecuencias y pequeños voltajes a valores digitales que posteriormente son nuevamente transformados en un tono de voz que oímos con una claridad y nitidez aceptable. No obstante muchos de estos dispositivos tienen al menos un teclado, una pantalla, un lector de tarjetas chip, un sistema de gestión de la batería, entre otros. Estas labores, como es de imaginarse, son mucho más sencillas y por ende no se justifica gastar tiempo valioso del DSP en la gestión de estas labores y es ese un escenario propicio para los microcontroladores. No es para sorprenderse encontrar teléfonos celulares que cuentan en su interior con 2 grandes chips, un DSP y un microcontrolador, cada uno de ellos haciendo su labor acorde a sus capacidades.

Un último caso de electrónica programable son los sistemas controladores proporcional integral y derivativo (PID). Estos dispositivos manipulan complejos sistemas y modelos de control, modelos los cuales incluyen la realización de integrales y derivadas (de ahí su nombre), cuyos resultados mezclados con los parámetros del modelo permi-

ten controlar un sistema y monitorear su correcto funcionamiento, tomando medidas correctivas en casos de fallas. Un caso muy común de uso de estos equipos es el control de temperatura, muchos escenarios reseñan situaciones en las cuales se necesita gobernar un muy importante y delicado sistema de temperatura en el cual las variaciones bruscas deben ser controladas ya sea por exceso o por defecto, eso sin contar que la velocidad de respuesta debe ser rápida y debe auto ajustarse ante la respuesta que dé el sistema controlado a los estímulos que genera el PID con el fin de gobernarlo, por lo tanto, él permanentemente revisa sus parámetros, medidas y se reajusta nuevamente para mantener en control el sistema y ese proceso de ajuste le demanda la evaluación de modelos matemáticos y la elección autónoma del mejor valor de ajuste para lograr una respuesta apropiada. La ventaja de este sistema radica en poder programarlo con el fin de ajustar correctamente los parámetros del modelo de control a emplear y como es habitual, muchos de ellos cuentan con IDE que son propias del fabricante del dispositivo, y cuenta con un juego de instrucciones que permite ajustarle sus modelos y parámetros.

Ejemplos innumerables se encuentran tanto en el mundo como en la electrónica y todos ellos apuntan a la construcción de aplicaciones óptimas sobre entornos de programación con juegos de instrucciones acordes a los dispositivos. En muchos de estos casos no basta con conocer de forma exhaustiva su construcción digital interna, o de manipular correctamente sus modelos matemáticos, sino que por el contrario, toda esta tecnología se desperdicia o se le da mal uso, cuando se construyen aplicaciones carentes de normas mínimas de algoritmia y principios fundamentales de programación. Es decir y para citar un ejemplo, por que resulta mejor un "for" o "para" con el fin de hacer un proceso repetitivo en comparación con un "while" o "mientras que"? La respuesta es tan larga como ancha, pero lo único cierto es que la incorrecta elección de uno de estos puede hacer la diferencia entre un software óptimo y uno de mala calidad.

III. Conclusiones

- La arquitectura de una máquina, su construcción y topología digital interna deben ir de la mano con las instrucciones dado el ancho de bits de los códigos de estas y el ancho de bits a procesar en su interior.
- El repertorio de instrucciones va de la mano con las capacidades internas del núcleo del microprocesa-

dor, estas deben ser coherentes con el sistema y deben ser óptimas con el funcionamiento del mismo.

- La optimización de cualquier instrucción permite una ejecución más rápida en el sentido de que cualquier paso de bits interno que se pueda realizar en forma autónoma, minimiza la cantidad de pasos para ejecutar una orden.
- Cada orden por compleja que sea, debe realizarse con el menor número de microinstrucciones para que estas sean lo más rápidas posibles.
- Un repertorio de instrucciones amplio, variado rico y eficaz, permite a los programadores la construcción de programas eficientes y mas rápidos, por lo tanto, la relación entre el software y el hardware es directa.

- La tendencia de la electrónica digital apunta cada vez mas a dispositivos micro-procesados con juegos de instrucciones más reducidos eficaces y por ende más programables.

IV. Referencias

- [1] Andrade, J.M. (2000). La Inteligencia Digital. *Revista Escuela Colombiana de Ingeniería*, 40, 32 – 39. Ed. Escuela Colombiana de Ingeniería. Bogotá.
- [2] Floyd, T. (2005). *Fundamentos de Sistemas Digitales* (7a Ed.), Madrid, España: Pearson Ed.
- [3] Hayes, J. (1998). *Organización y Arquitectura de Computadores* (3a Ed.), Boston, Massachussets, EE.UU.: Mc. Graw-Hill.
- [4] Mano, M. (1982). *Lógica Digital y Diseño de Computadores* (2da Ed.). Los Ángeles, California, EE.UU.: Prentice Hall.

Ángel Palacios Ingeniero Electrónico - ECI, Especialista en Telemática, Universidad de los Andes (UNIANDES). En la actualidad, es docente de tiempo completo del programa de Tecnología en Electrónica de la Corporación Universitaria Minuto de DIOS (UNIMINUTO).
apalacios@uniminuto.edu