**Procesador**

En ciencias de la computación, procesador se define como un circuito electrónico que ejecuta operaciones sobre información proveída externamente. Usualmente el término se refiere a la unidad central de procesamiento (CPU), aunque puede referirse también a la implementación de circuitos especializados para el manejo central de dispositivos como los SoC (*System on* *a Chip*).

El término 'microprocesador' simplemente hará alusión a un procesador

que cumple las funciones de un CPU en un solo chip.

**Bus**

Conjunto de elementos que facilitan la transmisión de datos entre los componentes de una computadora o fuera de ella. Es parte del hardware, que se forma con cables, pistas y dispositivos físicos.

Bus de control: dicta el acceso de datos y direcciones, ejecutando los mecanismos para evitar zonas de colisión. Maneja señales de órdenes y sincronización.

Bus de direcciones: en él se maneja la dirección de memoria a la que corresponde la información en el bus de datos, dado que cada segmento de memoria posee una dirección para su fácil acceso.

Bus de datos: permite la transmisión de información entre el CPU y los demás dispositivos.

**Memoria**

Dispositivo de almacenamiento de información (datos e instrucciones).

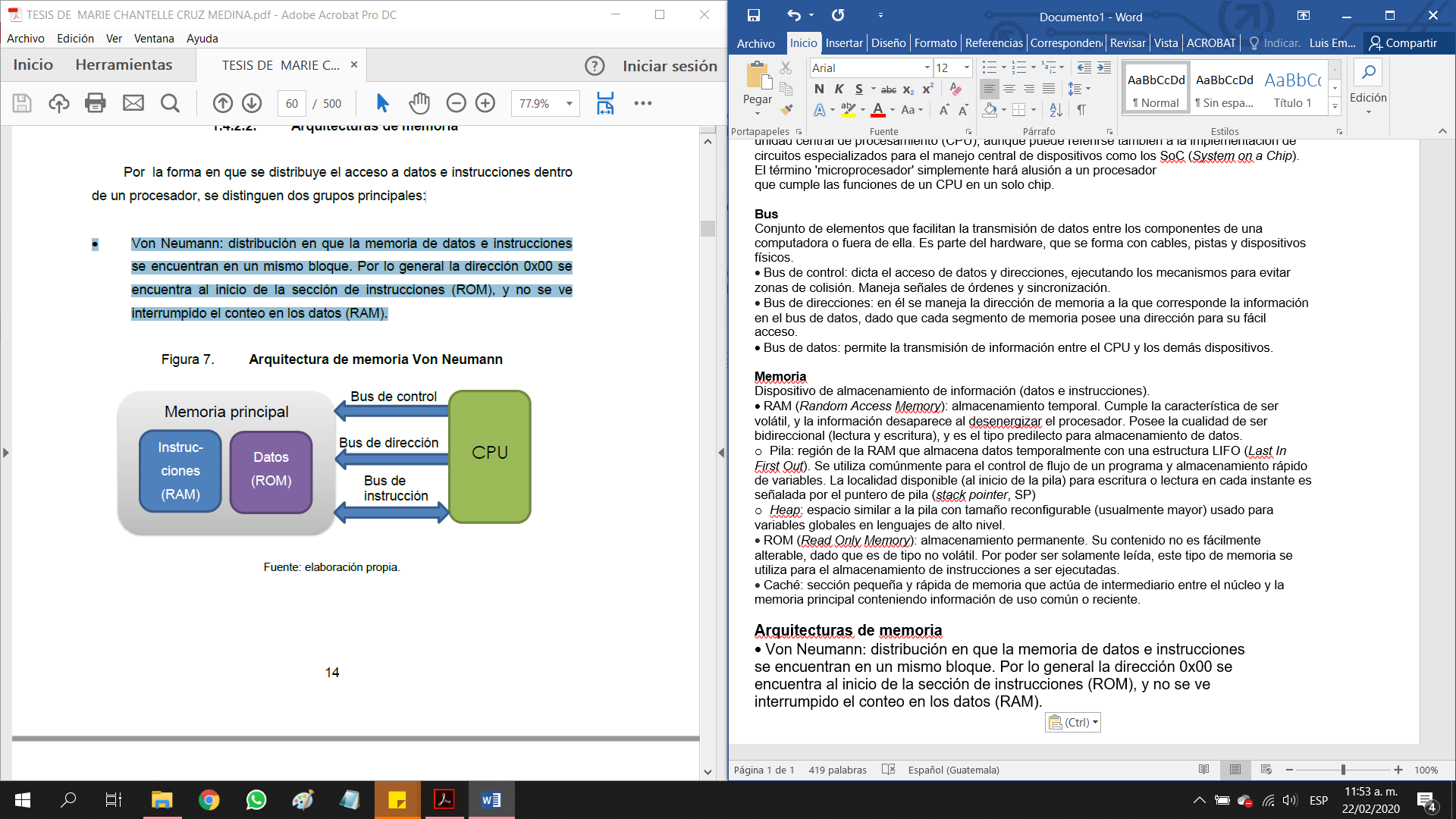
RAM (*Random Access Memory*): almacenamiento temporal. Cumple la característica de ser volátil, y la información desaparece al desenergizar el procesador. Posee la cualidad de ser bidireccional (lectura y escritura), y es el tipo predilecto para almacenamiento de datos.

o Pila: región de la RAM que almacena datos temporalmente con una estructura LIFO (*Last In First Out*). Se utiliza comúnmente para el control de flujo de un programa y almacenamiento rápido de variables. La localidad disponible (al inicio de la pila) para escritura o lectura en cada instante es señalada por el puntero de pila (*stack pointer*, SP)

o *Heap*: espacio similar a la pila con tamaño reconfigurable (usualmente mayor) usado para variables globales en lenguajes de alto nivel.

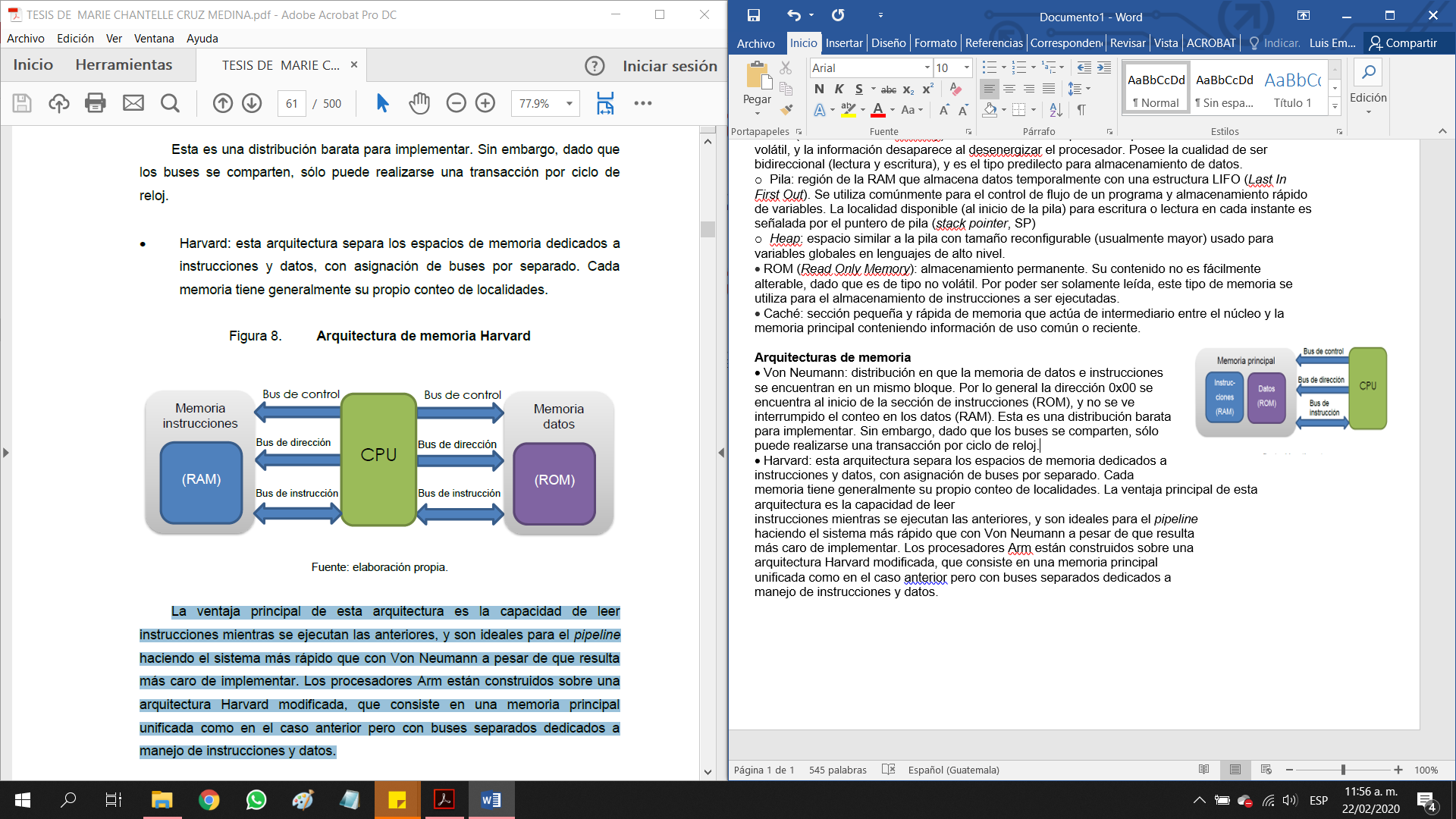
ROM (*Read Only Memory*): almacenamiento permanente. Su contenido no es fácilmente alterable, dado que es de tipo no volátil. Por poder ser solamente leída, este tipo de memoria se utiliza para el almacenamiento de instrucciones a ser ejecutadas.

Caché: sección pequeña y rápida de memoria que actúa de intermediario entre el núcleo y la memoria principal conteniendo información de uso común o reciente.



**Arquitecturas de memoria**

Von Neumann: distribución en que la memoria de datos e instrucciones se encuentran en un mismo bloque. Por lo general la dirección 0x00 se encuentra al inicio de la sección de instrucciones (ROM), y no se ve interrumpido el conteo en los datos (RAM). Esta es una distribución barata para implementar. Sin embargo, dado que los buses se comparten, sólo puede realizarse una transacción por ciclo de reloj.

Harvard: esta arquitectura separa los espacios de memoria dedicados a

instrucciones y datos, con asignación de buses por separado. Cada

memoria tiene generalmente su propio conteo de localidades. La ventaja principal de esta arquitectura es la capacidad de leer instrucciones mientras se ejecutan las anteriores, y son ideales para el *pipeline* haciendo el sistema más rápido que con Von Neumann a pesar de que resulta más caro de implementar. Los procesadores Arm están construidos sobre una

arquitectura Harvard modificada, que consiste en una memoria principal

unificada como en el caso anterior, pero con buses separados dedicados a

manejo de instrucciones y datos.

**Registro**

Memoria de poca capacidad y alta velocidad que permite un almacenamiento temporal para acceso rápido de información. Los registros usualmente se miden por su capacidad en bits, siendo estas implementaciones de *flip flops*.

Registros de datos

Registros de memoria

Registros de propósito general

Contador de programa

Registros de punto flotante

Registros constantes

Registros de propósito específico (puntero de pila, de estado)

Registros de banderas

**ALU (Unidad Lógico-Aritmética)**

Dispositivo digital capaz de ejecutar operaciones aritméticas, lógicas y es una parte fundamental del procesamiento.

Suma aritmética

Resta aritmética (complemento a 2)

Operaciones lógicas

Producto

Suma lógica

Comparación

Complementación

Enmascaramiento

Transferencia

Rotación

**Arquitectura de set de instrucciones (ISA)**

**RISC (*Reduced Instruction Set Computing*)**

Set de instrucciones diseñado para ejecuciones rápidas. Se caracteriza por contener instrucciones simples que se ejecuten en pocos ciclos de reloj (por lo general todas se ejecutan en la misma cantidad de ciclos), esto se debe a que todas son del mismo ancho de bits.

**CISC (*Complex Instruction Set Computing*)**

Set de instrucciones de instrucciones capaces de ejecutar tareas complejas. Al ser específicas, las instrucciones tienen varios anchos de bits en un mismo grupo, y el tiempo en que cada una se ejecuta también es variable. Los sets CISC contienen muchas más instrucciones que los RISC y por la disponibilidad de operaciones muy específicas se ha usado ampliamente para procesadores robustos, volviéndose base de varias otras ISA como x86 y x64.

***Pipeline***

Llamado también segmentación, es una técnica de paralelismo. La idea central del método es distribuir la carga del procesamiento de instrucciones en cada elemento del núcleo. Para utilizar *pipeline* el procesador cuenta con una sección dedicada que divide cada instrucción en pasos que pueden controlarse por distintas secciones al mismo tiempo. En el momento en que el sistema se estabiliza, se dice que se finalizará una instrucción por ciclo de reloj.

***Pipeline* de tres etapas**

La primera etapa es *Fetch*, que lee la instrucción desde memoria e incrementa el valor del registro de dirección de instrucción y el PC. La segunda etapa (*Decode*), decodifica la instrucción y prepara las señales de control para la ejecución. La tercera etapa (*Execute*), lee los operandos de los registros, realiza las operaciones de ALU, lee o escribe en memoria si es necesario y al final, escribe de vuelta los valores de registros modificados.

**Arquitectura Arm**

Siendo ARM una arquitectura RISC, posee características propias de este grupo como:

Una cantidad grande de registros

Modo de arquitectura l*oad/store,* donde las operaciones de procesamiento de datos se realizan sobre contenido en registros y no directamente en memoria.

Modos simples de direccionamiento con las direcciones necesarias para el *load/store* contenidas en registros.

Campos de instrucción de longitud fija, para simplificar la decodificación de instrucciones.

Además de estas características, ARM provee por su parte:

Control sobre ALU y desplazador de bits para maximizar su rendimiento en instrucciones de procesamiento de datos.

Modos de direccionamiento de auto decremento y auto incremento para optimizar los ciclos.

Instrucciones *Load and Store Multiple* para maximizar la salida de datos.

**Registros de propósito general**

La arquitectura tiene 31 registros de 32 bits (una palabra), de ancho cada uno. Sólo 16 de estos son visibles en el modo de usuario mientras el resto se utilizan en procesamiento. Dentro de los 16 registros accesibles (con nomenclatura Rn), se encuentran algunos con funciones especiales:

Puntero de pila: usualmente R13, lleva por siglas SP (*Stack Pointer*). Sirve para insertar y obtener información de la pila.

Registro de enlace: R14, con siglas LR (*Link Register*). Mantiene la dirección de la siguiente instrucción después de un salto en el programa.

Contador de programa: R15, con siglas PC (*Program Counter*). Indicador para apuntar a la instrucción dos líneas debajo de la ejecutada en el momento.

**Registros de estado**

Todos los demás registros del procesador, además de los de propósito general, están incluidos en este grupo. A su vez, los registros de estado se dividen en dos:

*Current Program Status Register* (CPSR): almacena las condiciones del estado actual del procesador. Contiene:

o Cuatro banderas de condición (N – negativo, Z – cero, C – acarreo, V - desbordamiento).

o Una *sticky flag* (Q). Indica si ha ocurrido saturación en instrucciones de aritmética saturada o un desbordamiento con signo en instrucciones de multiplicación acumulada.

o Cuatro banderas GE (*Greater than or Equal*). Indica si los resultados de operaciones con signo son no negativas o si una operación sin signo ha producido un acarreo.

o Dos bits de deshabilitación de interrupciones (I para IRQ y F para FIQ).

o Una máscara de bits (A), para modo de aborto por imprecisión.

o Cinco bits de codificación del estado actual del procesador. Este campo de modo funciona como máscara de bits para especificar una de las siguientes opciones:

o En caso de escribir un número no especificado, el procesador entra en un modo irrecuperable haciendo necesario el *reset*.

o Dos bits para determinar si se ejecutan *opcodes* de instrucciones ARM, Thumb (T) o Jazelle (J).

o Un bit de control de *endianness* (E) para operaciones load/store.

*Saved Program Status Register* (SPSR): almacena las condiciones del estado actual del procesador inmediatamente después de ocurrida una excepción. Es una forma de preservar el CPSR para su uso posterior.

**Modos de procesador**

Usuario: único no privilegiado, en él corren por defecto las aplicaciones

FIQ: entrada de interrupción de alta prioridad (*fast interrupt*)

IRQ: entrada de interrupción de baja prioridad (*normal interrupt*)

Supervisor (SVC): utilizado en un *reset* o en la presencia de interrupción por software.

Aborto: usado para manejar violaciones al acceso de memoria.

Indefinido: inducido para manejar instrucciones indefinidas.

Sistema: modo privilegiado con el uso de los mismos registros que en el modo de usuario.