

Título del reporte no muy largo

Nombre de la UEA

Alumno 1

Alumno 2

Alumno 3

Universidad Autónoma Metropolitana – Azcapotzalco

{alumno1, alumno2, alumno3}@correo.azc.uam.mx

7 de diciembre de 2016

Resumen

En esta sección el alumno redacta con un máximo de 50 palabras el contenido de la práctica: el problema resuelto, las restricciones que se requieren para obtener los resultados y los resultados principales. La primera parte de este apartado se redacta en presente y la presentación de resultados en pasado.

Palabras clave: reporte, formato, redacción

1. Introducción

Aquí se explica el conocimiento necesario para que el lector entienda el contenido de la práctica sin la necesidad de revisar otras fuentes bibliográficas. La lista de conceptos que se desarrollan en esta sección incluyen lo siguiente: marco contextual, definiciones básicas, la explicación de algoritmos clásicos, fundamentos teóricos, trabajo relacionado, el problema a resolver, la solución del problema y la estructura del reporte.

Para dar una idea al alumno sobre cómo desarrollar los párrafos que constituyen la introducción, a continuación se muestra una estrategia para describir el marco contextual. El marco contextual muestra al lector la relevancia temporal, social o intelectual del problema que se trata en el reporte. Por ejemplo, si se realiza una práctica sobre un procesador de propósito general se puede enunciar algo así: los procesadores de propósito general tienen una remarcada influencia en el trabajo productivo del hombre. Cotidianamente hacemos uso de sus prestaciones en computadoras personales, servidores, tabletas electrónicas y teléfonos inteligentes, por citar los más comunes. Empleando estos dispositivos podemos ejecutar una gran variedad de software que nos permite realizar desde operaciones aritméticas simples hasta el análisis de la interacción de los cuerpos celestes de la vía láctea. Lo mismo se aplica al desarrollo de los conceptos restantes. Siendo así, el alumno puede presentar la definición de un procesador de propósito general, el tipo de arquitectura que tiene, su filosofía de diseño, el conjunto de instrucciones que ejecuta y otras cosas más. Lo importante es que quede bien claro todo lo necesario para entender el contenido medular del reporte.

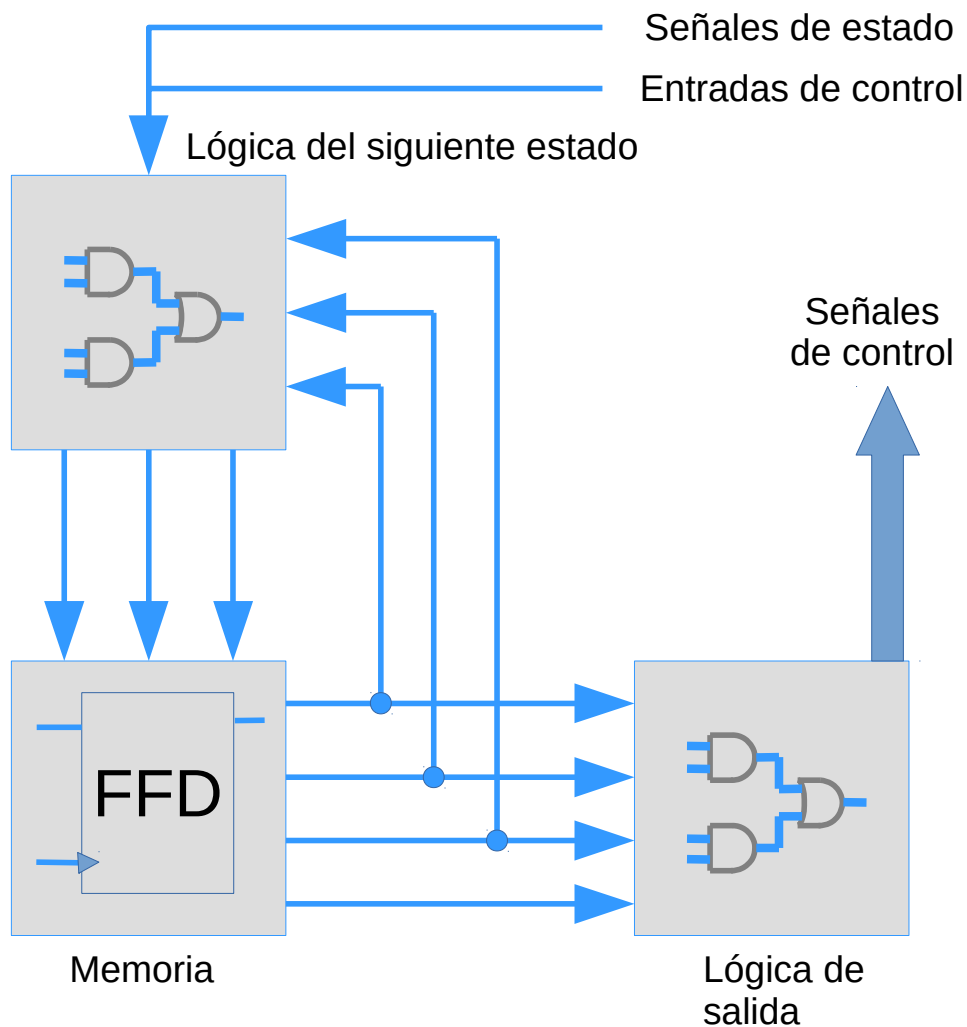


Figura 1: Máquina de estados finitos.

En la introducción y a lo largo del reporte es común el uso de elementos gráficos tales como figuras, gráficas o tablas. Todos estos tienen que tener una referencia y su correspondiente descripción, *e.g.* ., en la figura 1 se muestra el diagrama a bloques de una máquina de estados finitos (FSM). Los bloques que constituyen dicha máquina son la lógica combinacional del siguiente estado (lcse), una memoria y la lógica combinacional de salida. Como se muestra en la figura, la entrada del bloque lcse es el estado de la memoria y algunas señales de control mientras que su salida se conecta a la memoria de la máquina. La memoria de la máquina almacena su estado actual, el cual corresponde a un código binario determinado por el diseñador. La entrada de la lógica combinacional de salida es el estado actual de la máquina. Nota que la lógica combinacional de salida no tiene por entrada ninguna señal del exterior, por lo tanto, la FSM mostrada en la figura 1 corresponde a una máquina de Moore.

Dos párrafos que merecen especial atención en la introducción son los que describen el problema a resolver y los resultados obtenidos. El párrafo indica clara y explícitamente qué se está resolviendo. Suponiendo que se reporta el diseño de un procesador de propósito general, entonces la primera oración con la que inicia el párrafo del problema podría ser: en este escrito se reporta la estrategia de diseño seguida para implementar un procesador RISC que implementa en buena medida el ISA

del procesador ARC. Después, tendrían que venir algunas oraciones que den más detalle sobre cómo se diseña el procesador y las condiciones que se asumieron para ello. El último párrafo debe decir directamente qué resultado se obtuvo, p. ej., siguiendo las estrategias desarrollamos un procesador de propósito general de 16 bits que, a diferencia del ARC, ejecuta cualquier instrucción en cinco pasos.

La introducción finaliza con la presentación de la estructura del documento. Generalmente, la estructura es la siguiente: en la sección 2 se presentan las herramientas de hardware o software empleadas en la resolución del reporte. En la sección 3 se muestra la metodología seguida para resolver el problema. En la sección 4 se presentan y discuten los resultados y, finalmente, en la sección 5 se concluye este reporte.

2. Herramientas

En este apartado se menciona el software o hardware empleado en la realización de la práctica, por ejemplo:

1. visual code
2. iverilog
3. gtkwave
4. tarjeta de desarrollo Altera DE2

Enseguida se describe el software, se menciona la versión y se menciona su funcionamiento, poniendo especial énfasis en la explicación de la utilidad específica para llevar a cabo la práctica, p. ej., para escribir el código que describe nuestro procesador empleamos *visual code*, el cual es un editor de texto inteligente desarrollado por Microsoft que permite el uso de extensiones que facilitan la escritura de programas. En nuestro caso empleamos VerilogHDL.

Aunque simple, esta sección es muy importante porque da parte de la información que el lector necesita para reproducir el experimento. En el ejemplo que estamos desarrollando quizás no resulte de importancia el editor de texto empleado para escribir el código, pero sí es altamente importante la versión de iverilog y la tarjeta de desarrollo porque sin esos detalles es muy difícil obtener los resultados que se muestran en el reporte.

3. Metodología

Es aquí donde el alumno describe meticulosamente el procedimiento seguido para realizar la práctica. Esta parte es extremadamente importante porque su correcta descripción permite la reproducción del experimento y de los resultados. La sección debe tener el detalle necesario para que el lector pueda reproducir el experimento, así que se recomienda no escatimar en hojas.

El tiempo verbal con el que se escribe esta sección es el presente. Al alumno le puede resultar extraño el uso de ese tiempo porque él ya hizo los experimentos. No obstante, es importante aclarar que la metodología describe una serie de pasos inmutable que debe seguirse rigurosamente para obtener los resultados presentados en el reporte, siendo así, es más natural el tiempo presente que el pasado. El siguiente párrafo muestra un ejemplo de esto.

Como toda máquina de estados finitos, el diseño de nuestra unidad de control requiere que el diagrama de estados que representa a la FSM para implementar por inspección la lógica del siguiente estado mediante la técnica de codificación *one hot*. Además, se requiere conocer la arquitectura del conjunto de instrucciones y del formato de la palabra de control que maneja a la ruta de datos para diseñar la memoria y la lógica combinacional de salida.

Un vicio común encontrado en los reportes es la inclusión de código en la metodología. Se recomienda ampliamente evitar dicha práctica a menos que sea muy necesario. Esta recomendación parte del hecho de que el código depende de un lenguaje de programación que puede ser desconocido para el lector. Por esto se aconseja el uso de pseudocódigo en lugar de código ya que el primero puede ser implementado en cualquier otro lenguaje de programación. Un ejemplo de pseudocódigo se muestra en el algoritmo 1, el cual corresponde al cálculo del factorial de x . Si no se puede evitar la inclusión de una código, entonces éste presentarse como se muestra en el código 1.

Algoritmo 1 Cálculo recursivo del factorial de x

```

1: procedure FACTORIAL( $x$ )                                ▷  $x!$ 
2:   if  $x \leq 1$  then
3:      $n \leftarrow x$ 
4:   else
5:      $n \leftarrow x \cdot \text{FACTORIAL}(x - 1)$ 
6:   end if
7:   return  $n$ 
8: end procedure

```

Código 1: Implementación del algoritmo recursivo del cálculo del factorial de x en c++.

```

1  #include <iostream>
2  #include <cstdlib>
3
4  unsigned int factorial(unsigned int);
5
6  int main(int argc, char* argv[])
7  {
8      unsigned num = std::atoi(argv[1]);
9      std::cout << num << "! = " << factorial(num) << std::endl;
10 }
11
12 unsigned int factorial(unsigned int x)
13 {
14     unsigned int n;
15     if(x <= 1)
16         n = 1;
17     else
18         n = x * factorial(x - 1);
19     return n;
20 }

```

3.1. Diseño del diagrama de estados

En la metodología cabe lugar el uso de subsecciones, principalmente cuando un experimento se realiza en varias fases que requieren su propia metodología. En el caso del diseño del procesador, puede emplearse una subsección para describir el diseño de cada componente del procesador y otra para describir el banco de pruebas para evaluar su funcionamiento. Se recomienda evitar el uso de subsecciones cuando solo se tiene una en mente.

3.2. Diseño de la lógica combinacional del siguiente estado

3.3. Diseño de la memoria

3.4. Diseño de la lógica combinacional

3.5. Banco de pruebas

4. Resultados

En esta sección se presentan los resultados que se obtienen al emplear las herramientas presentadas en el apartado 2 y siguiendo la metodología presentada en la sección 3. El tiempo verbal que se usa en esta parte es el presente. Generalmente, la extensión de este apartado resulta breve si las dos secciones anteriores están bien escritas [1]. Aquí sólo hay que mostrar el resultado del problema. Siguiendo con el ejemplo del procesador, aquí se muestran el funcionamiento del procesador, *e.g.*, que ejecuta cualquier instrucción en cinco ciclos de reloj y que lo hace correctamente.

Comúnmente, en esta sección se usan gráfica y tablas para mostrar los resultados. Como se menciona en la introducción, estos elementos tienen que tener una referencia y además un nombre, *e.g.*, la tabla 1 muestra una tabla cualquiera que el alumno puede usar como base para mostrar sus resultados o configuraciones. Nota que el título de la tabla se encuentra en su parte superior mientras que el de las figuras está en la inferior. Es importante indicar que el uso de tablas y gráficas está justificado si facilita la explicación de los resultados. No vale la pena usar una gráfica o una tabla para mostrar los valores de una señal constante porque para eso basta una ecuación, *e.g.*, $v_{out} = 5\text{ V}$. No obstante, es muy recomendable incluir la gráfica que corresponde a la salida de la máquina de estados con respecto a una instrucción y el estatus del procesador.

Es muy importante discutir el porqué de los resultados, particularmente cuando el resultado es muy interesante. Aquí el alumno tiene que hacer un esfuerzo intelectual para explicar tanto lo esperado como lo no esperado, *e.g.*, si se mide el retardo de salida de un circuito combinacional propuesto en la literatura contra un diseño original, entonces se tiene que explicar por qué el diseño original es más rápido que el que se encuentra en el literatura.

5. Conclusión

Esta sección contiene un breve resumen escrito en tiempo pasado de lo hecho en la práctica. Además, aquí se resumen los resultados presentados en la sección anterior y se responden las preguntas planteadas en el experimento (si las hay). Finalmente, el alumno propone aplicaciones del hardware o

Tabla 1: Ejemplo de tabla.

| Name | | |
|------------|-----------|-------|
| First name | Last Name | Grade |
| John | Doe | 7.5 |
| Richard | Miles | 2 |

software, ventajas y desventajas con respecto a otros enfoques e ideas para mejorarlo. En nuestro caso cerramos este documento mencionando detalles generales sobre la redacción de un reporte.

Redactar un reporte es una tarea compleja que sólo puede dominarse mediante la práctica. Sin embargo, su redacción se facilita si se conoce la estructura del reporte y la de los párrafos que los conforman. Una manera de comenzar un párrafo consiste en escribir una idea y después apoyar esa idea con otras sentencias para finalmente concluir el párrafo mencionando cómo es que las ideas de apoyo soportan la idea principal. Como recordatorio, el formato de una oración en español se representa con la terna $\langle \text{sujeto}, \text{verbo}, \text{predicado} \rangle$. A la estructura anterior hay que agregarle los conectores necesarios para entrelazar las sentencias del párrafo. Si se tiene en cuenta esto, la tarea de redacción de un reporte se reduce a la redacción de párrafos y esto a su vez a la redacción de sentencias. Siendo así, puede declararse que la escritura de un reporte puede hacerse siguiendo la estrategia *divide y vencerás*.

En un reporte es necesario que el alumno cite todo aquello que no es materia intelectual suya. En efecto, no todas las ideas presentadas en un reporte provienen del intelecto del alumno. De hecho, mucho del conocimiento proviene de diversas fuentes bibliográficas, *e.g.*, libros, revistas científicas, páginas *web* y otras más. Todo conocimiento ajeno debe ser propiamente citado. Si se extrae un texto *verbatim* de una fuente, tal texto va entrecomillado y al final se muestra una referencia a la fuente original, *e.g.*, si se presenta una definición que proviene de otra fuente se hace lo siguiente: En [2] se indica que «un procesador es a menudo referido como la unidad de procesamiento central (CPU). La CPU simplemente es un microprocesador dedicado que sólo ejecuta instrucciones de software». Si sólo se usa una idea ajena o se quiere reforzar una idea propia con una ajena, entonces la sentencia donde se presenta la idea lleva una referencia hacia la fuente original, *e.g.*, un procesador de propósito general es un dispositivo de cómputo constituido por una máquina de estados de Mealey conectada a una ruta de datos [2, 3, 4]. Se recomienda que el alumno trate de generar sus propias ideas.

El estilo bibliográfico que se manejará en el curso es el de la IEEE escrito en español. Empleando LaTeX no es necesario preocuparse por el formato de las referencias ya que BibTeX se encarga de ello. Sólo es necesario que el alumno provee una base de datos en un archivo con extensión .bib que contenga la información bibliográfica. En este proyecto se encuentran dos archivos .bib, uno llamado `architecture.bib` y otro llamado `IEEEexample.bib`. Los dos son ejemplos que el alumno puede emplear para generar su propia bibliografía. Consulte [5] para mayor información sobre el estilo bibliográfico de la IEEE.

Aunque esta plantilla presenta detalles que son propios de la UEA de arquitectura de computadoras, la estructura puede emplearse para reportar experimentos de cualquier otra disciplina: fundamentos de redes de computadoras, comunicaciones digitales o análisis de algoritmos, por citar algunos. El autor del contenido de este material agradece cualquier observación, sugerencia o argumentación sobre esta plantilla.

Referencias

1. R. A. Day, *Cómo escribir y publicar artículos científicos*, ser. Publicación Científica y Técnica No. 598. Phoenix, EUA: The Oryx Press, 2005.
2. E. O. Hwang, *Digital Logic and Microprocessor design with VHDL*. California, EUA: Brooks/Cole 2005, 2005.
3. M. J. Murdocca y V. P. Heuring, *Computer Architecture and Organization: An Integrated Approach*, 2.^a ed. Nueva York, EUA: John Wiley & Sons, Inc., 2007.
4. W. Stallings, *Computer Organization and Architecture: Designing for Performance*, 9.^a ed. Nueva Jersey, EUA: Prentice-Hall, Inc., 2013.
5. P. Parada. (2014, Oct.) Estilo de referencias IEEE. [En línea]. Disponible: <http://www.derecho.unam.mx/integridad-academica/pdf/IEEE.pdf>