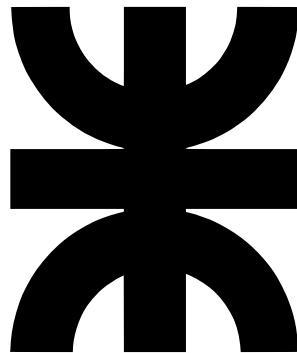


# Plataforma de Hardware Reconfigurable



**Luis Alberto Guanuco**

Departamento de Ingeniería Electrónica  
Universidad Tecnológica Nacional – Facultad Regional Córdoba

Este trabajo otorga el grado de  
*Ingeniero Electrónico*



A toda mi familia y quienes aportaron desinteresadamente en mi crecimiento personal.



## Agreadecimientos

Se agradece principalmente a la comunidad de *Hardware y Software Libre* ya que sin el aporte cooperativo y desinteresado de ellos sería imposible acceder a un sin fin de herramientas informáticas que son de gran importancia en el desarrollo territorial de cualquier Comunidad. También a los investigadores del *Centro Universitario de Desarrollo en Automoción y Robótica (CUDAR)*, quienes con su basta experiencia han brindando soporte técnico de gran manera el desarrollo del proyecto. Agradecemos el apoyo recibido por el *Departamento de Ingeniería Electrónica* de la *Universidad Tecnológica Nacional – Facultad Regional Córdoba*. Como así también a el *Laboratorio de Técnicas Digitales e Informática (LTDI)* por participar del proyecto, ofreciendo sus recursos humanos y físicos para la recepción del proyecto. A la *Agencia para el Desarrollo Económico de la ciudad de Córdoba (ADEC)* quienes nos han permitido participar de su programa *Córdoba Innovadora - Desarrollo Territorial en el Área Metropolitana de Córdoba*. En donde se ha podido adquirir experiencia en el manejo de proyecto y sobre todo la posibilidad de solventar económicamente el proyecto para su transferencia al LTDI. Y por último agradecer a todas las personas que en el transcurso del desarrollo han participado institucionalmente como así también en forma desinteresada.



## Resumen

El proyecto denominado *Plataforma de Hardware Reconfigurable*, ofrecer los recursos de *hardware* y *software* necesarios en el diseño de sistemas digitales reconfigurables de alta complejidad. El objetivo principal del proyecto es su implementación en el ámbito educativo, pero su aplicación puede extenderse a la industria ya que posee flexibilidad y recursos variados para la generación de diseños prototipos.

El desarrollo se encuentra publicado bajo condiciones legales que permite el *uso* y *modificación* de todo el proyecto por cualquier diseñador que así lo desee, lo que asegura la total *libertad* a quienes se estén formando en el diseño de sistemas electrónicos digitales. La Plataforma de *Hardware* reconfigurable ofrece un desarrollo alternativo en las plataformas digitales comerciales ya que se encuentre adaptada a las necesidades requeridas por el ámbito educativo regional. En el desarrollo han participado activamente docentes e investigadores, quienes aportaron información que en parte han definido el perfil técnico y académico del proyecto.





# Contenidos

Contenidos	ix
Lista de Figuras	xiii
Lista de Tablas	xv
Nomenclatura	xviii
<b>1 Introducción</b>	<b>1</b>
1.1 Motivación y objetivos . . . . .	1
1.2 Transferencia del proyecto . . . . .	2
1.3 <i>Hardware &amp; Software</i> . . . . .	2
1.4 PHR (Plataforma de Hardware Reconfigurable) . . . . .	3
1.5 Licencia Libre . . . . .	4
<b>2 Fundamentos y Planteamiento del Desarrollo</b>	<b>5</b>
2.1 Dispositivos Lógicos Programables . . . . .	5
2.1.1 FPGAs . . . . .	7
2.2 Ámbito de implementación . . . . .	8
2.2.1 Antecedentes . . . . .	8
<b>3 Antecedentes</b>	<b>11</b>
3.1 Placa MiniLab . . . . .	11
3.2 Kit CPLD . . . . .	11
3.3 OT-CPLD . . . . .	11
<b>4 Costos y Financiamiento del proyecto</b>	<b>13</b>
4.1 Planificación de los gastos . . . . .	13
4.2 Procedimientos en adquisición de materias primas . . . . .	13

---

4.3	Financiación del Proyecto . . . . .	13
4.4	¿Costos en <i>Software</i> ? . . . . .	13
<b>5</b>	<b>Implementación del Proyecto</b>	<b>15</b>
5.1	Necesidad de plataformas de <i>hardware</i> . . . . .	15
5.2	Antecedentes . . . . .	15
5.3	Cátedras beneficiadas . . . . .	15
5.4	Inserción de los lenguajes descriptivos . . . . .	15
<b>6</b>	<b>El proyecto PHR</b>	<b>17</b>
6.1	Desarrollos de referencias . . . . .	17
6.2	Estructura general del proyecto . . . . .	17
6.3	Definición de estructura de las placas . . . . .	17
6.4	Selección de dispositivos principales . . . . .	17
6.5	Descripción de las placas . . . . .	17
6.5.1	PHR . . . . .	17
6.5.2	S3Power . . . . .	17
6.5.3	OOCDLink . . . . .	17
<b>7</b>	<b>Proyectos <i>Open Hardware</i></b>	<b>19</b>
7.1	Fundamentos y principios básicos . . . . .	19
7.2	Tipos de Licencias . . . . .	19
7.3	Importancia en la Educación . . . . .	19
7.4	Algunos proyectos <i>Libres</i> . . . . .	19
7.4.1	Nacionales . . . . .	19
7.4.2	Internacionales . . . . .	19
<b>8</b>	<b>My Third Chapter</b>	<b>21</b>
8.1	First Section of the Third Chapter . . . . .	21
8.1.1	First Subsection in the First Section . . . . .	21
8.1.2	Second Subsection in the First Section . . . . .	21
8.1.3	Third Subsection in the First Section . . . . .	21
8.2	Second Section of the Third Chapter . . . . .	22
	<b>Referencias</b>	<b>23</b>

Contenidos	xi
<hr/>	
<b>Anexo A Dispositivos Electrónicos</b>	<b>25</b>
A.1 Evolución de los Circuitos Integrados . . . . .	25
<b>Anexo B Installing the CUED Class file</b>	<b>27</b>



# Lista de Figuras

2.1	Arquitectura básica de una FPGA . . . . .	7
2.2	Diferentes <i>package</i> de las FPGAs comerciales. . . . .	8
2.3	Best Animations . . . . .	10
A.1	Evolución y comparación de los circuitos integrados . . . . .	26



# Lista de Tablas

2.1	Evolución de los PLDs . . . . .	6
2.2	Características de FPGAs fabricadas por Xilinx. . . . .	9
2.3	Características de FPGAs fabricadas por Actel. . . . .	9
8.1	Table with Borders . . . . .	22





# Nomenclatura

## Acrónimos / Abreviaturas

3D-IC *Three-Dimensional Integrated Circuit*

CLB *Configurable Logic Block*

CPLD *Complex Programmable Logic Device*

DLL *Delay-Locked Loop*

DSP *Digital Signal Processor*

EEPROM *Electrically Erasable Programmable Read-Only Memory*

F2F *Face-to-face*, en microelectrónica, F2F representa la disposición “cara a cara” de las uniones entre capas del diseño en la tecnología 3D-IC

FF flip-flop, circuito que tiene dos estados estables y puede ser usado para almacenar información

FPGA *Field Programmable Gate Array*

GNU *GNU's Not Unix*

HDL *Hardware Description Language*

IC *Integrated Circuit*

JTAG *Joint Test Action Group*

PALCE *PAL CMOS Electrically erasable/programmable*

PAL *Programmable Array Logic*

PCI *Peripheral Component Interconnect*

PHR *Plataforma de Hardware Reconfigurable*

PLD *Programmable Logic Device*

PLL *Phase-Locked Loop*

SEM *Scanning Electron Microscope*

SRAM *Static-RAM*

TSV *Through Silicon Via*

# Capítulo 1

## Introducción

### 1.1 Motivación y objetivos

En el proceso de aprendizaje de las denominadas *Técnicas Digitales* se tiene un eslabón importante que es la implementación de estos sistemas a la práctica (*hardware*). Teorías como el *Álgebra de Bool* con operaciones digitales simples, hasta la síntesis de *filtros digitales* son prácticas comunes en la formación del profesional en el área de la Ingeniería Electrónica y resulta fundamental su ejercitación para concluir el ciclo de enseñanza. Actualmente existen nuevas herramientas al alcance de la mano que pueden facilitar en la formación de estudiante de ingeniería, sobre todo las denominadas *Plataformas o Kit Educativos* que cuentan con una enorme complejidad y recursos en su diseño pero no así para el usuario final. Situación tan distinta en décadas pasadas donde la industria e instituciones académicas de países desarrollados eran quienes contaba con dichas plataformas.

En el avance tecnológico exponencial que se dio en las últimas décadas del siglo XX se podrían destacar varios logros pero quizá el que toma gran relevancia es el acceso a la información (*Internet*). Esta herramienta permite que regiones en desarrollo puedan llevar adelante estudios de nuevas tecnologías, comparando lo dicho con años anteriores donde no se contaba con la masificación de la información. En la mayoría de los casos las instituciones académicas resultan llevar la bandera en estas búsquedas del conocimiento. En búsqueda de lograr nuestro objetivo principal, ofrecer una herramienta personalizado a las necesidades de los estudiantes que se *inician* en el área del diseño de sistemas digitales basados en Dispositivos Lógicos Programables (*Programmable Logic Devices*), se ha trabajado tomando como referencias proyectos universitarios en nuestra Facultad como también de otros países además de los perfiles

de diseño que tienen las placas comerciales.

## 1.2 Transferencia del proyecto

Como se dijo en la sección anterior, el objetivo del proyecto es diseñar una plataforma que pueda ser útil para los estudiantes iniciales en el diseño de sistemas digitales. En el análisis de la implementación del proyecto en el ámbito académico se optó en coordinar un trabajo conjunto con el *Departamento de Ingeniería Electrónica* y el *Laboratorio de Técnicas Digitales e Informática (LTDI)*, ambas instituciones pertenecientes a la *Universidad Tecnológica Nacional – Facultad Regional Córdoba*. El LTDI es el laboratorio informático principal con el que cuentan los estudiantes de Ingeniería Electrónica y cuenta con recursos físicos y humanos necesarios para cubrir la demanda de las distintas Cátedras que allí se dictan. Miembros del LTDI han formado parte de la generación de documentación necesaria para el proyecto y la intención a futuro es que dicho personal sea quienes definan mejoras/modificaciones a versiones futuras de la *Plataforma de Hardware Reconfigurable*.

## 1.3 Hardware & Software

Las herramientas de *hardware* y *software* son comunes en el campo laboral de un Ingeniero Electrónico. Si bien el profesional Electrónico puede ejercer su actividad en diferentes ámbitos industriales, siempre requerirá el conocimientos de diseño de sistemas físicos como también interactuar con programas informáticos. Seguramente el desenvolvimiento del Ingeniero Electrónico de décadas anteriores contaba con otras herramientas y lograba desarrollarse con éxito en su profesión, pues seguramente así lo requería la Industria. Hoy la situación es distinta y quizá exista mucha bibliografía que detalle con mayor claridad esta observación que aquí se hace, no es nuestra intención comparar, simplemente poner en contexto la importancia del vínculo entre el mundo físico y los sistemas informáticos.

El nombre del proyecto, *Plataforma de Hardware Reconfigurable*, hace referencia al diseño de placas electrónicas que presentan la posibilidad de reconfigurar su estructura interna y así sintetizar diferentes arquitecturas diseñadas por el usuario, aquí es donde toma importancia en definirla *Hardware Reconfigurable*. Pero a medida que se avance en la lectura del presente Informe se podrá notar la importancia que representa el *software* en el proyecto.

Actualmente en el mercado de las plataformas educativas de sistemas embebidos existe verdaderamente una enorme variedad de excelentes productos. Cada uno de estos desarrollos se encuentran orientados a un determinado grupo de usuarios pero la mayoría de estos diseños tienen recursos de *hardware* en común:

- Dispositivo principal de proceso o *hardware*
- Puerto de programación y depuración (*debugging*)
- Periféricos
- Herramientas de *software*

A la hora de determinar que plataforma se pretende adquirir se debe realizar un análisis de los requerimientos de la implementación y obviamente tener con que presupuesto se dispone. La mayoría de las plataformas comerciales son adquiridas con la finalidad de realizar *prototipos* que permitan clarificar y definir un desarrollo final funcional. Persiguiendo el mismo fin pero en el ámbito académico, el perfil de la plataforma debe ofrecer los recursos físicos estratégicamente necesarios para el avance tecnológico de la región. De esta forma permitirá que los profesionales formados puedan implementar nuevas tecnologías en la industria local.

## 1.4 PHR (Plataforma de Hardware Reconfigurable)

La plataforma PHR se presenta como una herramienta para las prácticas en las Cátedras del área de Técnicas Digitales. Su estructura está basada en la caracterización planteada en la Sección 1.3. Es decir, la PHR presenta básicamente un Dispositivo Lógico Programable (PLD, siglas en inglés) al cual se tendrá acceso para sintetizar arquitecturas digitales implementadas mediante el uso de Lenguajes Descriptivos de Hardware (HDL, siglas en inglés). Anexo a este dispositivo central se dispone de una memoria de programación donde se almacenará la arquitectura implementada. Estos se accede mediante un puerto de programación estándar denominado JTAG. Además se ofrece dispositivos periféricos que permiten comunicarse al exterior de la placa mediante un puerto de comunicación serial, pulsadores, llaves, indicadores LEDs y display de 7 segmentos. Toda la energía necesaria es proporcionada por una fuente de alimentación capaz de ofrecer tanto los niveles de potencia necesarios para el dispositivo central y sus periféricos, como así también controlar los requerimientos de encendido

del dispositivo central (cuestión que se verá en detalle en Capítulos posteriores). La principal herramienta de *software* que complementa a la plataforma PHR es el encargado de reconfigurar el dispositivo central mediante el puerto JTAG. Se presentaron varias alternativas de *software* que pudieran ser útiles en el proceso de programación tanto libres como privativos. Más adelante se describirá con detalles cada una de estas variantes y se justificará la seleccionada.

## 1.5 Licencia Libre

El proyecto se encuentra distribuido en con licencias Libres, para ser más exactos *GNU General Public License Version 3 (GPLv3)*. Lo que permitirá que cualquier persona pueda acceder a la documentación como así también los esquemáticos y códigos (*scripts*) que se ha desarrollado. La mayoría de las referencias utilizadas en el desarrollo de la plataforma PHR han sido de otros proyectos que cuentan con licencias similares a la GPLv3. Esta decisión se toma principalmente por una cuestión legal pero también para darle un valor y divulgación a estos diseños que tienen objetivos similares a lo que aquí se pretende lograr con la PHR. De esta manera el proyecto formará parte de otros diseños similares que se encuentran distribuidos en la comunidad electrónica en forma libre o abiertas. Si bien hay desarrollos similares en otras Universidades del mundo, se pretende incentivar a que los estudiantes de esta región no solo tengan acceso a esta tecnología, sino también que se atrevan a realizar modificaciones y aportes a la plataforma PHR.

# Capítulo 2

## Fundamentos y Planteamiento del Desarrollo

### 2.1 Dispositivos Lógicos Programables

Los *Dispositivos Lógicos Programables* (PLDs) fueron introducidos a mediados de 1970s. La idea era construir circuitos lógicos combinacionales que fueran *programables*. Contrariamente a los microprocesadores, los cuales pueden *correr* un programa sobre un hardware *fijo*, la programabilidad de los PLDs hace referencia a niveles de *hardware*. En otras palabras, un PLD es un chip de *propósitos generales* cuyo *hardware* puede ser reconfigurado dependiendo de especificaciones particulares del programador.

EL primer PLD se llamaba PAL (*Programmable Array Logic*). Estos dispositivos disponían solo de compuertas lógicas (no tenían flip-flop), por lo que solo permitía la implementación de circuitos *combinacionales*. Para salvar este problema, Los *registered* PLDs fueron lanzados pocos después, los cuales incluían un flip-flop por cada salida del circuito. Con esta versión de los PAL, se podría implementar funciones *secuenciales* simples.

En el comienzo de 1980s, se agregaba más circuitos lógicos adicionales a la salida de los PLD. Este circuito de salida se lo identificaba como celda, llamado también *Microcelda*, que contenía (además de flip-flop) compuertas lógicas y multiplexores. Por otra parte, la celda era reprogramable, permitiendo varios modos de operación. Además, se podía proveer una señal de retorno (*feedback*) desde la salida del circuito a la lógico principal de la PAL, lo que le daba mayor flexibilidad a estos dispositivos reprogramables. Esta nueva estructura era llamada *generic PAL* o GAL. Una arquitectura de dispositivo similar fue conocido como PALCE (*PAL CMOS Electrically*

*erasable/programmable*).

Todos estos chips (PAL, *registered* PLD, y GAL/PALCE) son ahora categorizados como SPLDs (*Simple* PLDs). Los dispositivos GAL/PALCE son los únicos fabricados aún en una encapsulado independiente.

Luego, varios dispositivos GAL fueron fabricados en un solo chip, usando un esquema de direccionamiento más sofisticado, mayor tecnología en su fabricación, y varias características adicionales (como soporte JTAG e interfaces para varios estándares lógicos). Esta nueva propuesta se la conoció como CPLD (*Complex* PLD). Los CPLDs son actualmente muy populares debido a su alta densidad, funcionalidad, y bajo costo.

Finalmente, a mediados de 1980s, las FPGA (*Field Programmable Gate Array*) fueron introducidos al mercado de los IC. Las FPGAs diferían de los CPLDs en su arquitectura, tecnología, recursos internos, y costo. Estos dispositivos tenían como principal objetivo su implementación en diseños de gran requerimientos en recursos de hardware como así también un alto rendimiento.

Un pequeño resumen de los diferentes dispositivos PLDs se puede observar en la Tabla 2.1.

PLDs	Sample PLD (SPLD)	PAL Registered PAL GAL
	Complex PLD (CPLD)	
	FPGA	

Tabla 2.1 Evolución de los PLDs

Por último, todos los PLDs (*simple* o *complex*) son no volátiles. Estos puede ser OTP (*One-Time Programmable*), en la que pequeños fusibles electrónicos son usados para la reprogramación, de igual forma que las EEPROM o memorias Flash. Las FPGAs, por otra lado, son en su mayoría volátiles. Para estas últimas se deben usar dispositivos externos para cargar las conexiones.

Actualmente los dispositivos más utilizados son los CPLDs y FPGAs. De estas dos tecnología, se optó por trabajar con las FPGAs pues, como bien se dijo antes, cuentan con mayor recursos de *hardware* que los CPLDs. Lo que permitirá dar un mayor margen a los desarrollos digitales que se quieran implementar. Se detallará con mayor profundidad la tecnología y estructura de las FPGAs.



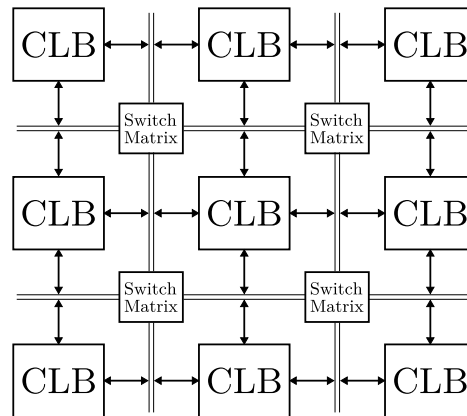


Figura 2.1 Arquitectura básica de una FPGA

### 2.1.1 FPGAs

Las FPGAs fueron introducidas al mercado por la empresa Xilinx Inc. a mediados de 1980s. Estos dispositivos se diferencian de los CPLDs en su arquitectura, tecnología de almacenamiento, funcionalidades integradas, y costo, y además están orientadas a la implementación de altos rendimientos y grandes tamaños en lo que se refiere a recursos de hardware.

La arquitectura básica de una FPGA se ilustra en la Figura 2.1. Esta consiste de una matriz de *CLBs* (*Configurable Logic Blocks*), interconectados por un arreglo de matrices de conmutadores (*Switch Matrix*). Para caracterizar con más detalle estos dispositivos se debe recurrir a la información de los fabricantes, donde además se puede disponer de un interfaz JTAG a diversos niveles lógicos, otra funcionalidad como memorias SRAM, multiplicadores de clock (PLL o DLL), interfaz PCI, etc. Algunos chips también incluyen bloques dedicados como multiplicadores, DPSs, y microprocesadores.

Las FPGAs pueden ser muy sofisticadas. La fabricación de chips con una tecnología CMOS de 90 nm., con nueve capas de cobre y más de 1000 pines de I/O, se encuentran actualmente disponibles en el mercado. Algunos ejemplos de los empaquetados (*package*) de las FPGAs son ilustrados en la Figura 2.2, en los cuales se puede apreciar uno de los *package* más pequeños (Fig. 2.2a) con 100 pines, un *package* de tamaño mediano (Fig. 2.2b) de 324 pines, y uno de los grandes *package* con 1156 pines (Fig. 2.2c).

Varias compañías fabrican FPGAs, como Xilinx., Actel, Altera, QuickLogic, Atmel, etc. Ejemplo de dos fabricantes (Xilinx y Actel) se disponen en las Tablas 2.2 y 2.3. Como puede verse, estos dispositivos pueden contener miles de flip-flops y varios

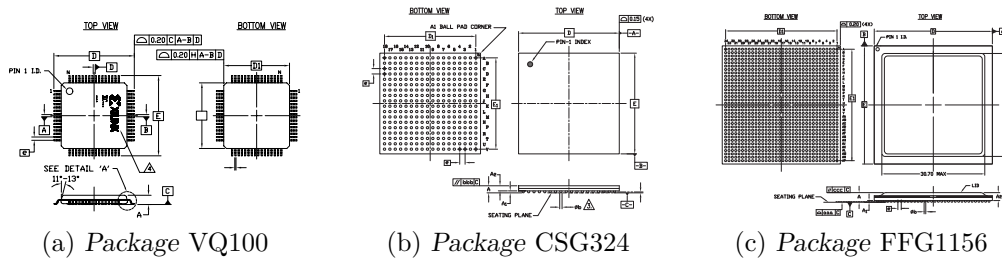


Figura 2.2 Diferentes *package* de las FPGAs comerciales.

millones de compuertas lógicas.

Nótese que todas las FPGAs de Xilinx usan SRAM para almacenar las interconexiones, por lo que son reprogramables, pero volátiles (es así que requieren de una ROM externa). en cambio, las FPGAs de Actel son no-volátiles (estos usan fusibles electrónicos), pero no son reprogramables (excepto una familia, la cual usa memoria *Flash*). Ya que cada enfoque tiene sus propias ventajas y desventajas, la aplicación real dictará cual arquitectura de chip es la apropiada.

## 2.2 Ámbito de implementación

### 2.2.1 Antecedentes

Familia	Virtex II Pro	Virtex II	Virtex E	Virtex	Spartan 3	Spartan IIE	Spartan II
CLBs	352 – 11.024	64 – 11.648	384 – 16.224	384 – 6.144	192 – 8.320	384 – 3.456	96 – 1.176
Celdas Lógicas	3.168 – 125.136	576 – 104.882	1.728 – 73.008	1.728 – 27.648	1.728 – 74.880	1.728 – 15.552	432 – 5.292
<i>System gates</i>		40k – 8M	72k – 4M	58k – 1.1M	50k – 5M	23k – 600k	15k – 200k
Pines de I/O	204 – 1200	88 – 1108	176 – 804	180 – 512	124 – 784	182 – 514	86 – 284
Flip-flops	2.816 – 88.192	512 – 93.184	1.392 – 64.896	1.392 – 24.576	1.536 – 66.560	1.536 – 13.824	384 – 4.704
Frec. máxima interna	547 MHz	420 MHz	240 MHz	200 MHz	326 MHz	200 MHz	200 MHz
Voltaje	1.5 V	1.5 V	1.8 V	2.5 V	1.2 V	1.8 V	2.5 V
Inter-conexiones	SRAM	SRAM	SRAM	SRAM	SRAM	SRAM	SRAM
Tecnología	0.13 $\mu\text{m}$ 9 capas de cobre CMOS	.15 $\mu\text{m}$ 8 capas de metal CMOS	0.18 $\mu\text{m}$ 6 capas de metal CMOS	0.22 $\mu\text{m}$ 5 capas de metal CMOS	0.09 $\mu\text{m}$ 8 capas de metal CMOS		
SRAM bits (Bloques de RAM)	216k – 8M	72k – 3M	64k – 832k	32k – 128k	72k – 1.8M	32k – 288k	16k – 56k

Tabla 2.2 Características de FPGAs fabricadas por Xilinx.

Familia	Accelerator	ProASIC	MX	SX	eX
Módulos lógicos	2.016 – 32.256	5.376 – 56.320	295 – 2.438	768 – 6.036	192 – 768
<i>System gates</i>	125k – 2M	75k – 1M	3k – 54k	12k – 108k	3k – 12k
Pines de I/O	168 – 684	204 – 712	57 – 202	130 – 360	84 – 132
Flip-flops	1.344 – 21.504	5.376 – 26.880	147 – 1.822	512 – 4.024	128 – 512
Frec. máxima interna	500 MHz	250 MHz	250 MHz	350 MHz	350 MHz
Voltaje	1.5 V	2.5 V, 3.3 V	3.3 V, 5 V	2.5 V, 3.3 V, 5 V	2.5 V, 3.3 V, 5 V
Inter-conexiones	<i>Antifuse</i>	<i>Flash</i>	<i>Antifuse</i>	<i>Antifuse</i>	<i>Antifuse</i>
Tecnología	0.15 $\mu\text{m}$ 7 capas de metal CMOS	.22 $\mu\text{m}$ 4 capas de metal CMOS	0.45 $\mu\text{m}$ 3 capas de metal CMOS	0.22 $\mu\text{m}$ CMOS	0.22 $\mu\text{m}$ CMOS
SRAM bits	29 k – 339 k	14 k – 198 k	2.56 k	n.a.	n.a.

Tabla 2.3 Características de FPGAs fabricadas por Actel.



Figura 2.3 Best Animations

## Subplots

I can cite Wall-E (see Fig. 2.3b) and Minions in despicable me (Fig. 2.3c) or I can cite the whole figure as Fig. A.1

# Capítulo 3

## Antecedentes

**3.1 Placa MiniLab**

**3.2 Kit CPLD**

**3.3 OT-CPLD**



# Capítulo 4

## Costos y Financiamiento del proyecto

4.1 Planificación de los gastos

4.2 Procedimientos en adquisición de materias primas

4.3 Financiación del Proyecto

4.4 ¿Costos en *Software*?





# Capítulo 5

## Implementación del Proyecto

5.1 Necesidad de plataformas de *hardware*

5.2 Antecedentes

5.3 Cátedras beneficiadas

5.4 Inserción de los lenguajes descriptivos



# Capítulo 6

## El proyecto PHR

6.1 Desarrollos de referencias

6.2 Estructura general del proyecto

6.3 Definición de estructura de las placas

6.4 Selección de dispositivos principales

6.5 Descripción de las placas

6.5.1 PHR

6.5.2 S3Power

6.5.3 OOCDLink



# Capítulo 7

## Proyectos *Open Hardware*

7.1 Fundamentos y principios básicos

7.2 Tipos de Licencias

7.3 Importancia en la Educación

7.4 Algunos proyectos *Libres*

7.4.1 Nacionales

7.4.2 Internacionales



# Capítulo 8

## My Third Chapter

### 8.1 First Section of the Third Chapter

And now I begin my third chapter here ...

And now to cite some more people Ancy et al. [1], Read [2]

#### 8.1.1 First Subsection in the First Section

...and some more

#### 8.1.2 Second Subsection in the First Section

...and some more ...

##### First subsub section in the second subsection

...and some more in the first subsub section otherwise it all looks the same doesnt it?  
well we can add some text to it ...

#### 8.1.3 Third Subsection in the First Section

...and some more ...

##### First subsub section in the third subsection

...and some more in the first subsub section otherwise it all looks the same doesnt it?  
well we can add some text to it and some more and some more and some more and

some more and some more and some more and some more ...

### **Second subsub section in the third subsection**

...and some more in the first subsub section otherwise it all looks the same doesn't it? well we can add some text to it ...

## **8.2 Second Section of the Third Chapter**

and here I write more ...

Now we can refer to the table using Table. 8.1.

Tabla 8.1 Table with Borders

1	2	3
4	5	6
7	8	9



# Referencias

- [1] Ancey, C., Coussot, P., and Evesque, P. (1996). Examination of the possibility of a fluid-mechanics treatment of dense granular flows. *Mechanics of Cohesive-frictional Materials*, 1(4):385–403.
- [2] Read, C. J. (1985). A solution to the invariant subspace problem on the space  $l_1$ . *Bull. London Math. Soc.*, 17:305–317.



# Anexo A

## Dispositivos Electrónicos

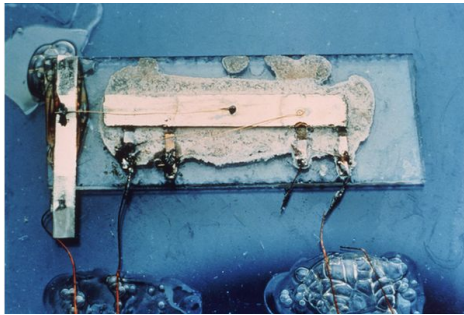
### A.1 Evolución de los Circuitos Integrados

El costo total de un sistema digital se puede decir que está determinado por las placas electrónicas, sistema de alimentación, interconexiones y encapsulado, así como el diseño, testeo, y otros costos de producción que también fueron comentados en la Sección 1.3 del Capítulo introductorio. Aquí los dispositivos digitales representan una fracción del total del costo del sistema. Consecuentemente, todo diseñador pretende realizar un sistema donde sea mínimo el total de circuitos integrados utilizados, lo que en muchos casos permitirá reducir las dimensiones de la placa electrónica, requerimientos de potencia y otros costos relacionados.

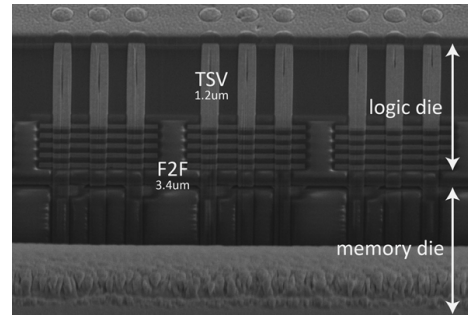
En las últimas décadas, el número de transistores que forman un circuito integrado (IC, por sus siglas en inglés) se ha incrementado debido al creciente nivel de integración en los procesos de fabricación de estos dispositivos electrónicos. La primera aparición pública del IC<sup>1</sup> fue en Mayo de 1952 por Geoffrey W. A. Dummer, quién presentó la idea de la integración en un Simposio sobre “Avances en Componentes Electrónicos de Calidad”. En el año 1959 Jack Kilby fue quién diseñó el primer circuito integrado, dispositivo de germanio que *integraba* seis transistores en una misma base semiconductor para formar un oscilador de rotación de fase. Y es así que en la década de '80 se manipulaba aproximadamente un millón de transistores, a fines de la misma década billones de transistores que integran un IC. Actualmente se está trabajando en tecnologías mucho más novedosas, sobre todo en el proceso de fabricación, los Circuitos Integrados en 3 Dimensiones (3D-IC). Aquí se trabaja con dos o más capas

---

<sup>1</sup>Se conoce que Werner Jacobi, en 1949, tramitó la patente de un circuito integrado pero no se registra que se haya realizado una implementación funcional.



(a) El primer IC creado por Jack Kilby en 1958. Contiene solo un transistor y componentes de soporte sobre una base de germanio.



(b) Imagen microscópica (SEM) de un 3D-IC desarrollado por la empresa Tezzaron Semiconductor. Se identifica una parte del chip, lógica, memoria, vías de conexión verticales (TSV) y terminales de unión (F2F).

Figura A.1 Evolución y comparación de los circuitos integrados

de componentes electrónicos activos que son integrados tanto en forma vertical como horizontal sobre un mismo circuito.

# Anexo B

## Installing the CUED Class file

$\LaTeX$ .cls files can be accessed system-wide when they are placed in the  $\langle\text{texmf}\rangle/\text{tex}/\text{latex}$  directory, where  $\langle\text{texmf}\rangle$  is the root directory of the user's  $\TeX$  installation. On systems that have a local texmf tree ( $\langle\text{texmflocal}\rangle$ ), which may be named texmf-local or localtexmf, it may be advisable to install packages in  $\langle\text{texmflocal}\rangle$ , rather than  $\langle\text{texmf}\rangle$  as the contents of the former, unlike that of the latter, are preserved after the  $\LaTeX$  system is reinstalled and/or upgraded.

It is recommended that the user create a subdirectory  $\langle\text{texmf}\rangle/\text{tex}/\text{latex}/\text{CUED}$  for all CUED related  $\LaTeX$  class and package files. On some  $\LaTeX$  systems, the directory look-up tables will need to be refreshed after making additions or deletions to the system files. For  $\TeX$ Live systems this is accomplished via executing texhash as root.  $\text{MIK}\TeX$  users can run initexmf -u to accomplish the same thing.

Users not willing or able to install the files system-wide can install them in their personal directories, but will then have to provide the path (full or relative) in addition to the filename when referring to them in  $\LaTeX$ .

