

**Universidad Autónoma de Querétaro**

**Facultad de Ingeniería**  
**Ingeniería en Automatización**

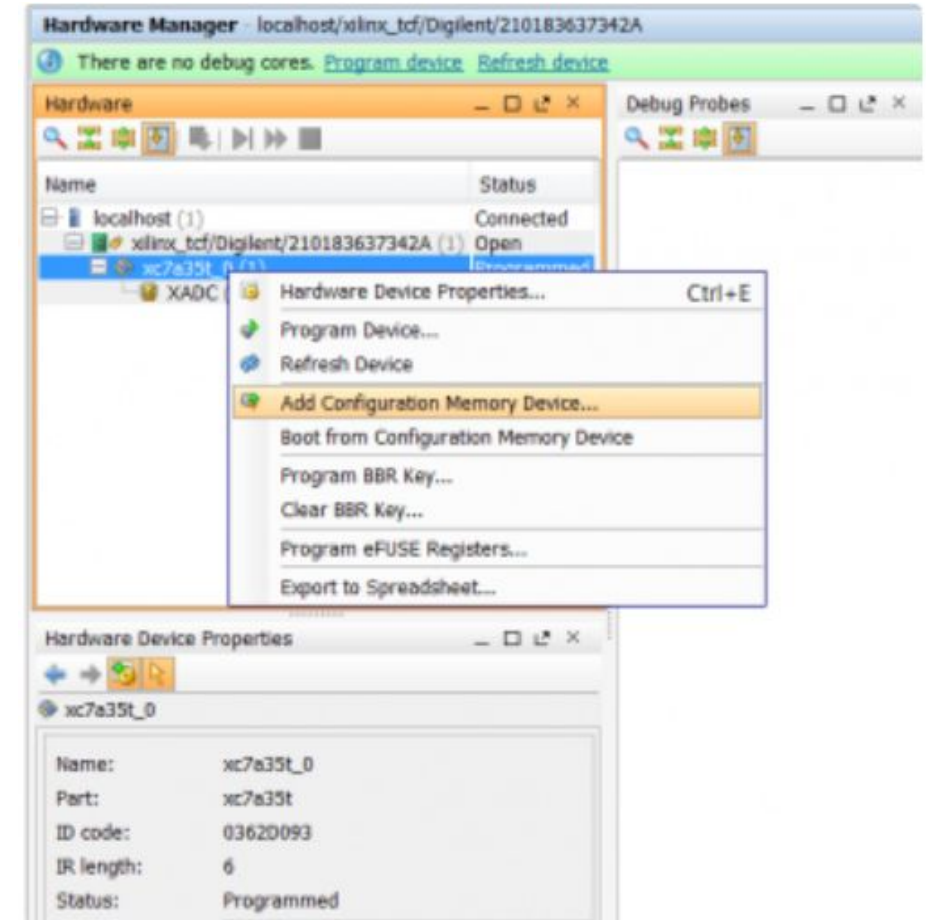


# ADC

**M en C. Marcos Romo Avilés**

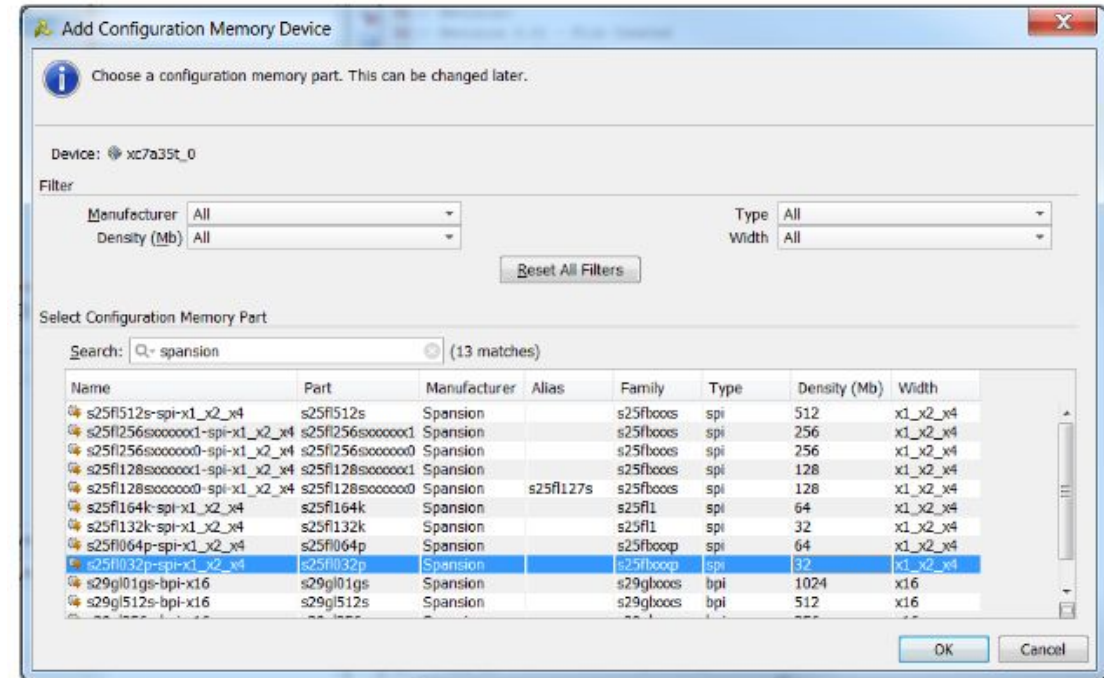
# Practica 11: Vivado

- Quad SPI Flash es una memoria no volátil que el chip FPGA de Basys3 analiza cada vez que se inicia.
- Si se flashea Quad SPI, el FPGA se programará a sí mismo con el contenido que se encuentra en la memoria flash de Quad SPI.
- Este método de programación de la placa es ideal cuando se tiene un proyecto final que se desea demostrar o mostrar y que no necesita ser editado y, por lo tanto, reprogramado.
  1. Asegúrese de que el puente en JP1 esté en la posición QSPI
  2. En la ventana del Administrador de hardware, en Hardware, haga clic derecho en su dispositivo y haga clic en Agregar dispositivo de memoria de configuración.



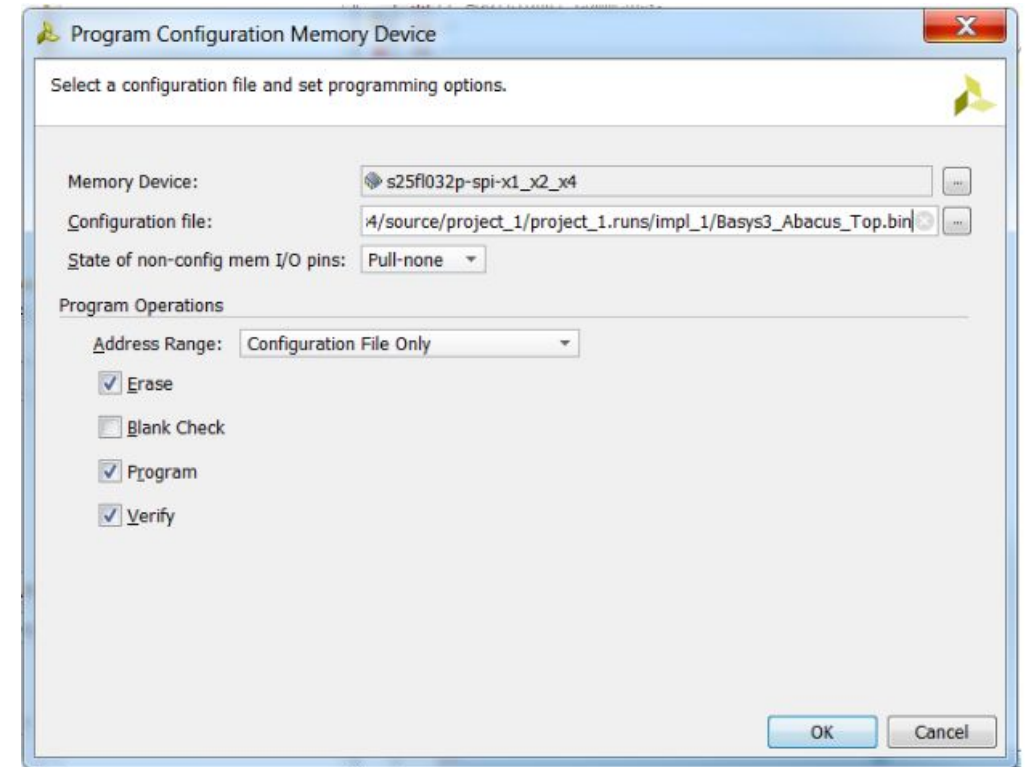
# Practica 11: Vivado

3. Aparecerá esta ventana. Busque “Spansion” y seleccione el dispositivo de 32 bits (resaltado a continuación).
4. Haga clic en Aceptar en la siguiente ventana que le preguntará si desea programar el dispositivo de memoria de configuración.



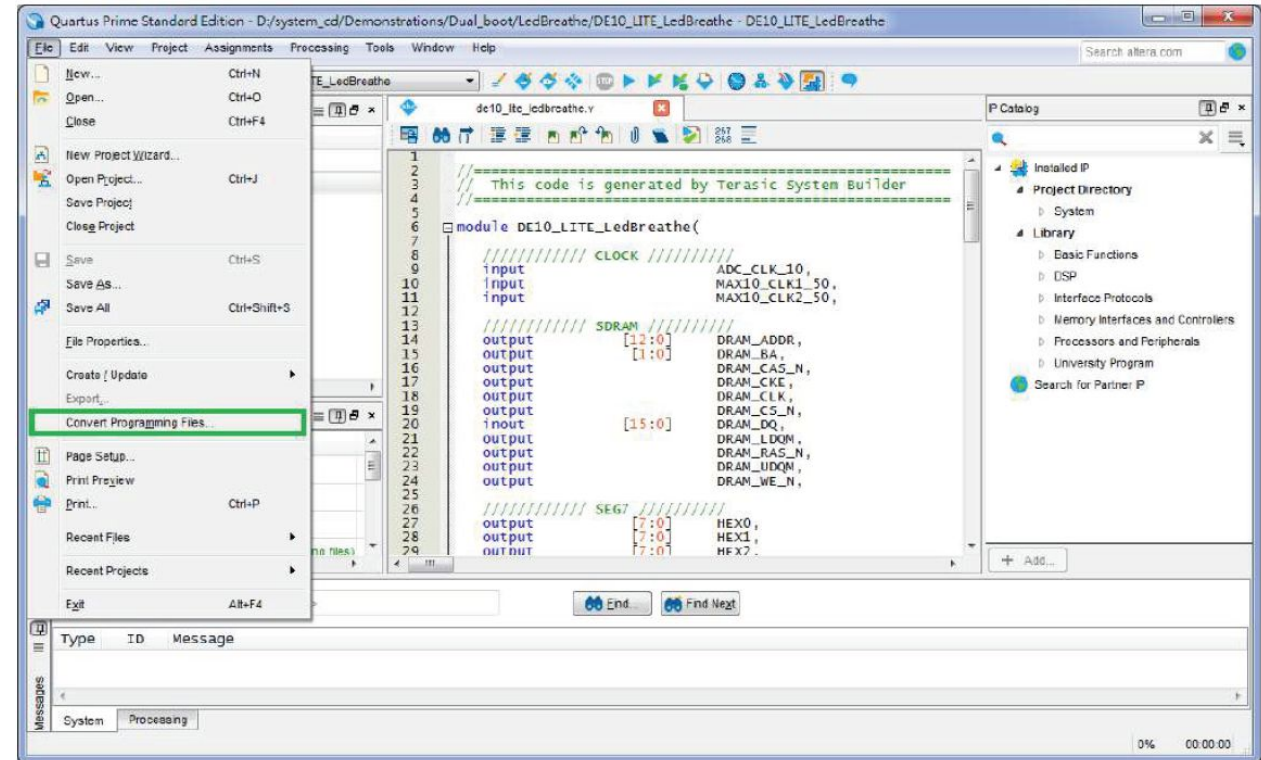
# Practica 11: Vivado

5. Seleccione el archivo .bin donde pide un archivo de configuración y finalmente haga clic en Aceptar.
- Nota: Vivado borrará ahora el archivo de configuración anterior y reprogramará su Basys3 con el archivo de demostración. A partir de ahora, cuando encienda el Basys3, la demostración se cargará al inicio hasta que lo re programe.



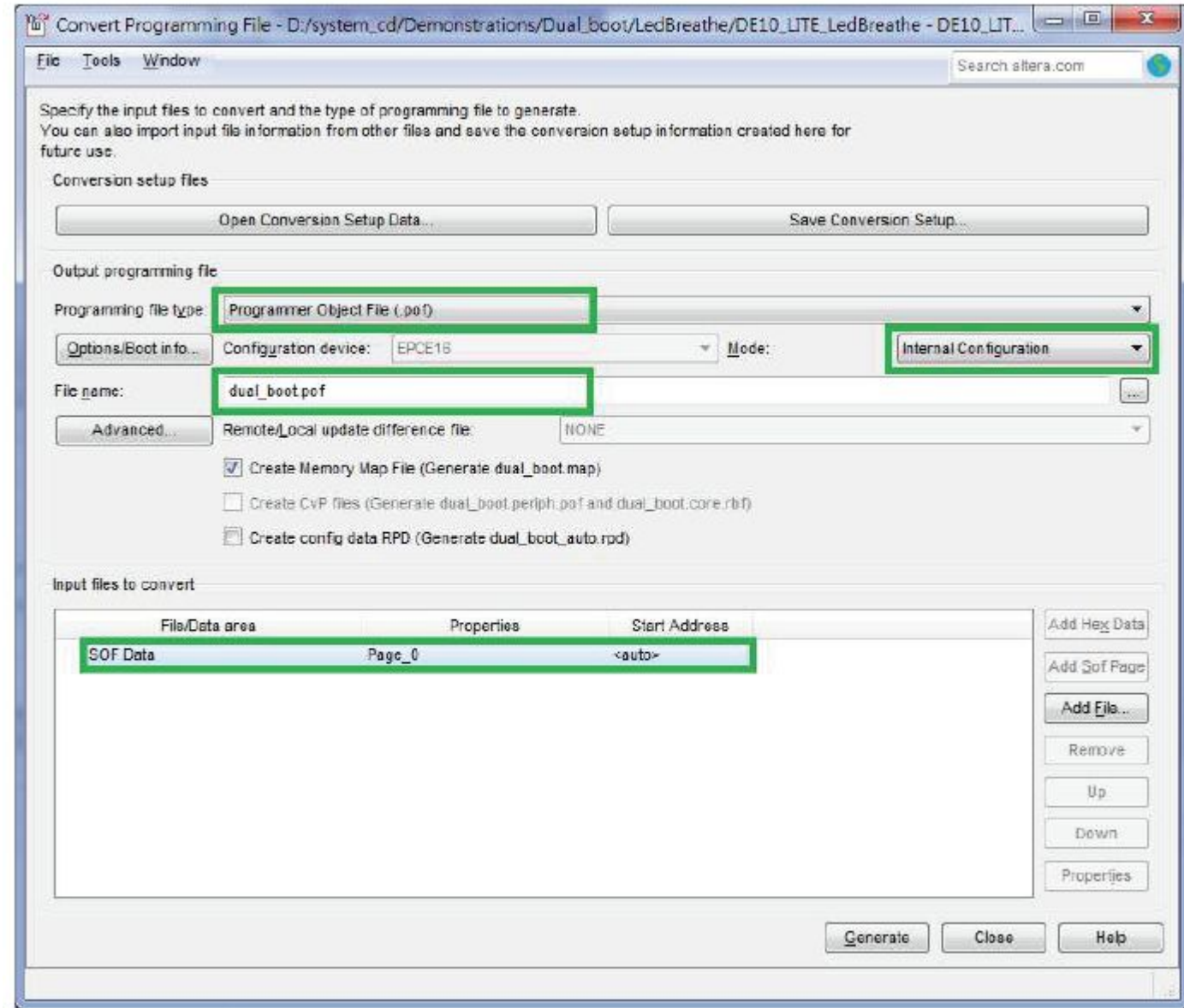
# Practica 11: Altera

- Los datos de configuración que se escribirán en CFM formarán parte del archivo de objeto del programador (.pof).
  - Estos datos de configuración se cargan automáticamente desde CFM a los dispositivos MAX 10 cuando se enciende la placa.
1. Seleccione Convertir archivos de programación en el menú Archivo de Quartus II para abrir una nueva ventana, como se muestra en la Figura.



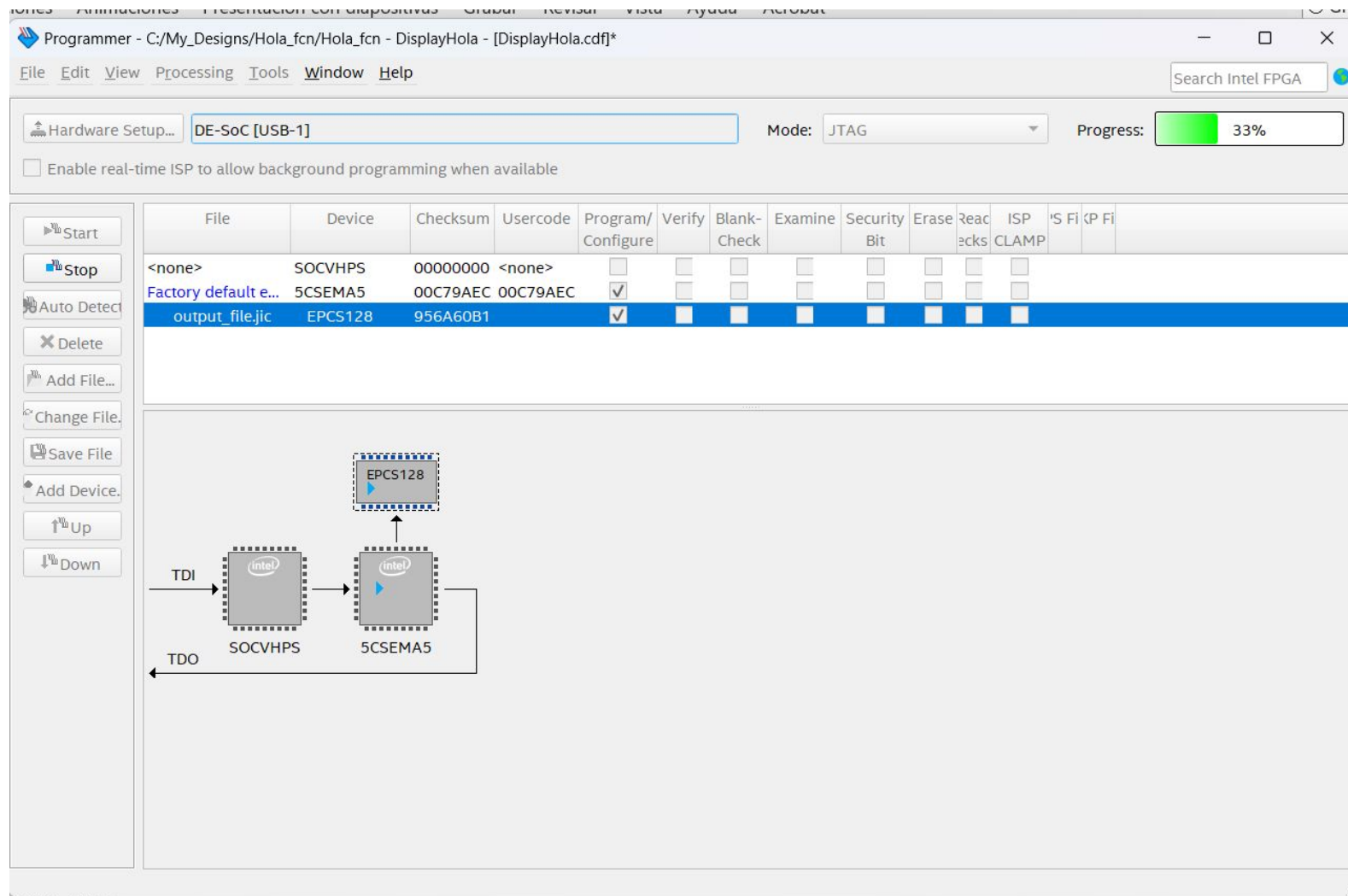
# Practica 11: Altera

- Haga clic en Agregar archivo y seleccione .sof para que sean los datos sof de Page\_0.
- Click Generate.



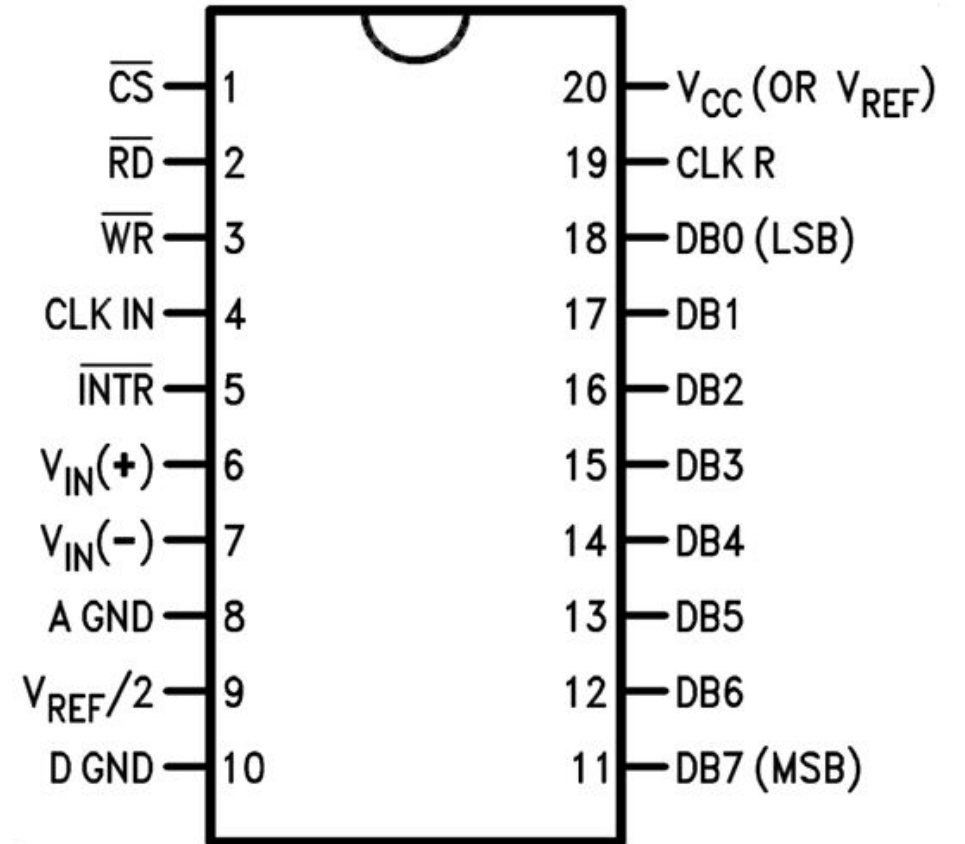


# Practica 11: Altera



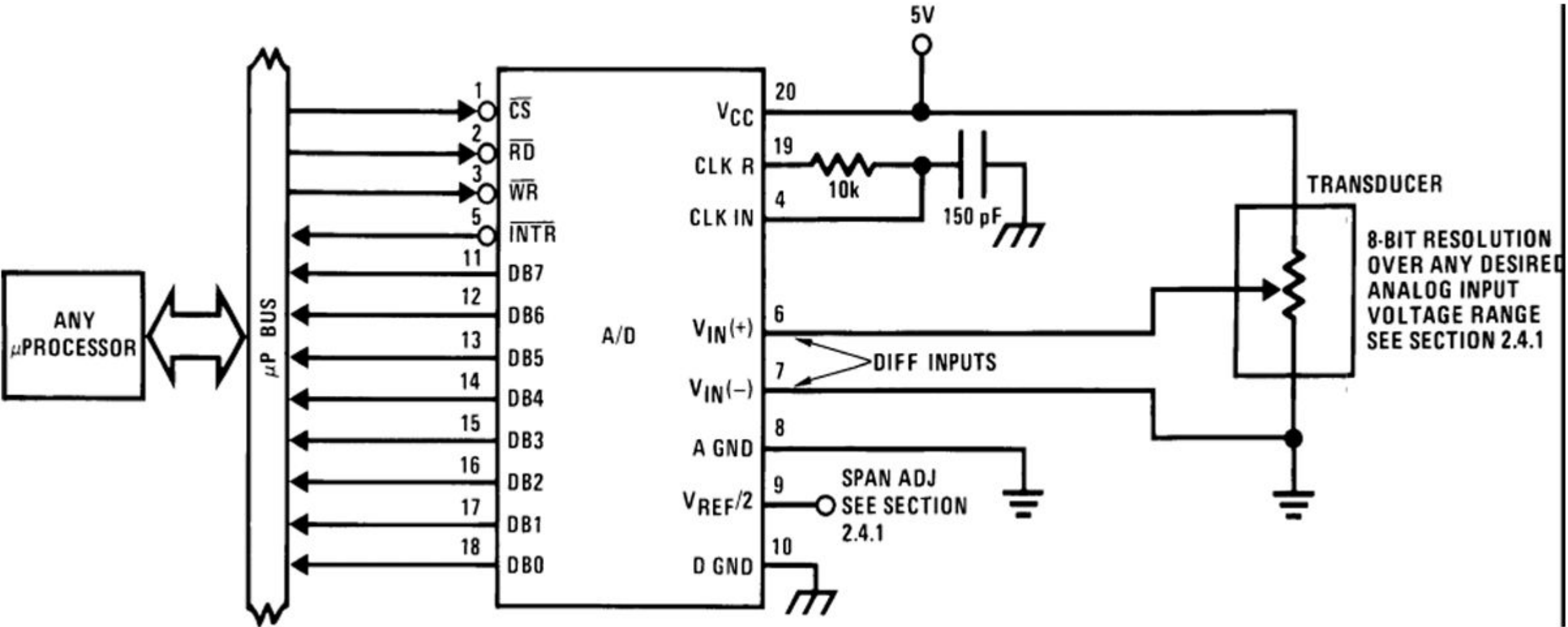
# Practica 11: ADC

PIN		I/O	DESCRIPTION
NO.	NAME		
1	CS	I	Chip Select
2	RD	I	Read
3	WR	I	Write
4	CLK IN	I	External Clock input or use internal clock gen with external RC elements
5	INTR	O	Interrupt request
6	$V_{IN}(+)$	I	Differential analog input+
7	$V_{IN}(-)$	I	Differential analog input-
8	A GND	I	Analog ground pin
9	$V_{REF}/2$	I	Reference voltage input for adjustment to correct full scale reading
10	D GND	I	Digital ground pin
11	DB7	O	Data bit 7
12	DB6	O	Data bit 6
13	DB5	O	Data bit 5
14	DB4	O	Data bit 4
15	DB3	O	Data bit 3
16	DB2	O	Data bit 2
17	DB1	O	Data bit 1
18	DB0 (LSB)	O	Data bit 0
19	CLK R	I	RC timing resistor input pin for internal clock gen
20	$V_{CC}$ (or $V_{REF}$ )	I	+5V supply voltage, also upper reference input to the ladder



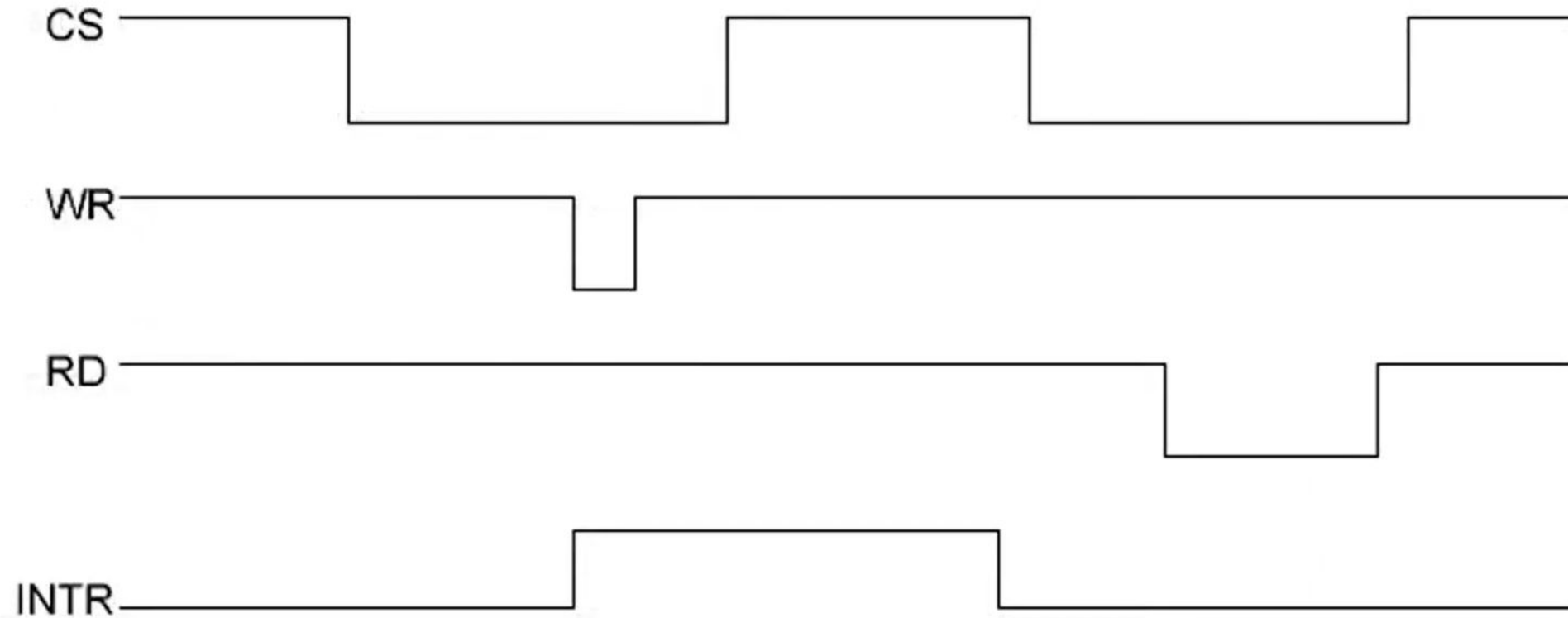


# Practica 11: ADC



- El dispositivo tiene tierras separadas para los voltajes Analógicos y digitales
- **CS (entrada)**
  - Cuando está en alto las salidas digitales entran en Hi-Z
  - Debe estar en bajo para que las entradas RD y WR tengan efecto.
- **RD (entrada)**
  - Cuando está en bajo las salidas digitales tienen los valores lógicos que representan la última conversión A/D.
  - Si está en alto las salidas tienen alta Hi-Z.
- **WR (entrada)**
  - Cuando se aplica un pulso en bajo a esta inicia una nueva conversión (de alto a bajo y de bajo a alto.)
- **INTR (salida)**
  - Se pone en bajo cuando finaliza una conversión y cambia a un nivel alto cuando inicia la conversión

# Practica 11: ADC



- Diseñar una máquina de estados para el control de ADC.
- No se permite reloj interno del ADC.
- No se permite modo libre.
- Configurar el correctamente el reloj.
- Se debe poder seleccionar cuando leer la salida y cuando convertir la entrada.
- Si no está seleccionado el modo de conversión se debe mantener la última conversión a la salida.
- La descripción se debe cargar a la memoria no volátil de la FPGA.