

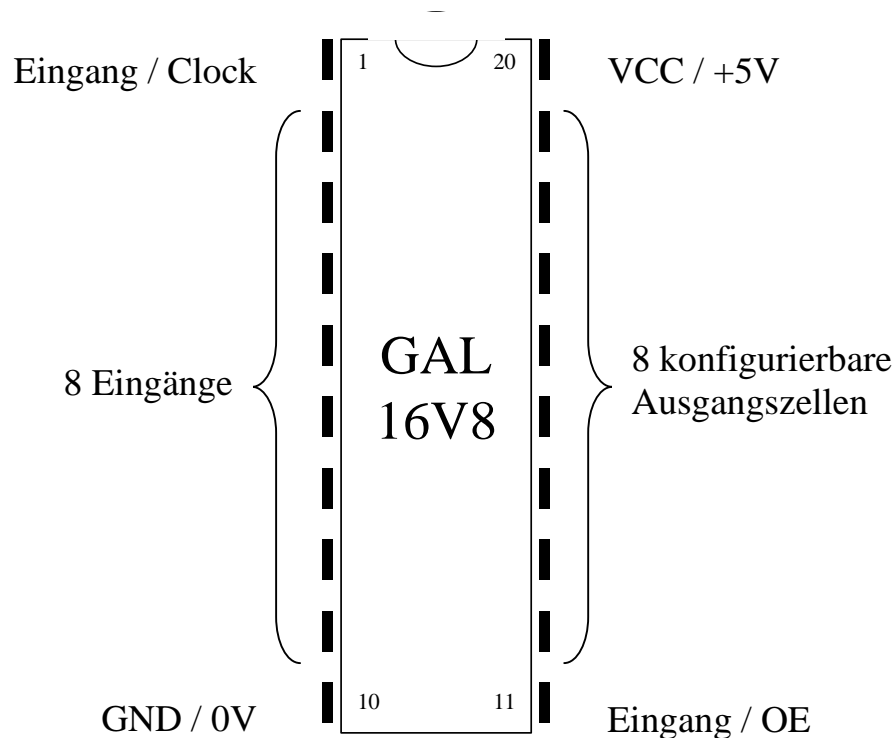
4. Laboreinheit - Hardwarepraktikum SS 2002

1. Versuch Programmierbare Logik

Am Beispiel des GAL16V8 und eines GAL Development Systems werden die Möglichkeiten und Einsatzgebiete von programmierbare Logikbausteine (PLDs) untersucht. Die GAL's werden auch als Generic Array Logic bezeichnet.

Vorbereitung

Wiederholen Sie die Grundlagen für die folgenden Grundschaltungen: Komparator, Multiplexer, Demultiplexer, Adressdecoder, Codeumsetzung. Als Zusatzliteratur wird empfohlen - Bitterle, Dieter: GALs programmierbare Logikbausteine in Theorie und Praxis (Franzis 1992, ISBN 3-7723-5903-5). Unter Anleitung wird Ihnen zu Beginn der Laboreinheit der Umgang mit dem GAL Developmentssystem und den unterschiedlichen Betriebsmodi der GALs vermittelt. Anhand einfacher Logikgrundschaltungen wird der Entwurf, die Simulation, die Programmierung und die hardwaremäßige Testung vorgestellt.



Der GAL16V8 besitzt 10 feste Eingänge und 8 konfigurierbare Ausgangszellen. Diese Ausgangszellen lassen sich auch als Eingang definieren bzw. können als Rückführung eines Signals in die Schaltung verwendet werden. Damit ist der GAL ein flexibel einsetzbarer Baustein mit 10 – 17 Eingängen und 8 – 1 Ausgängen. Der Pin 1 und der Pin 11 besitzen eine Sonderstellung beim GAL16V8, weil an diese Pins der Takt (CLOCK) bzw. der Output enable (OE) angelegt werden kann. Diese zwei Pins sind mit den internen D-FlipFlops verbunden und steuern somit die Registerausgänge des ICs.

Die Betriebsmodi des GALs

Kombinatorischer Ausgang

- Die Ausgänge sind ständig aktiv und es ist kein hochohmiger Zustand möglich
- Pro Ausgang sind maximal acht Produktterme mit ODER Verknüpfung realisierbar
- Ein Produktterm kann nur aus Eingängen gebildet werden
- Die Eingänge stehen auch invertiert zur Verfügung
- Die Ausgangssignale werden im GAL nicht zurückgekoppelt
- Die Ausgänge können wahlweise aktiv High oder Low definiert werden

Tristate Ausgang

Durch ein Steuersignal kann der Ausgang in einen Tristate Zustand gebracht werden. Das Steuersignal kann durch einen Produktterm gebildet werden.

- Die Ausgänge sind Tristate-Ausgänge
- Die Freigabe des Ausgangssignals erfolgt durch einen Produktterm
- Pro Ausgang können maximal sieben Produktterme ODER -verknüpft werden
- Produktterme können aus Eingängen und rückgekoppelten Ausgängen gebildet werden
- Sechs der acht Ausgänge werden ins GAL zurückgekoppelt
- Ein Ausgang kann wahlweise aktiv High oder Low definiert werden
- Die Eingänge und rückgekoppelten Ausgänge stehen im GAL auch invertiert zur Verfügung
- Ein Tristate-Ausgang mit Rückkopplung kann auch als Eingang verwendet werden

Register-Ausgang

- Mindestens ein Ausgang muss als Register-Ausgang definiert werden
- Ein Ausgang kann wahlweise als Register-Ausgang oder als Tristate-Ausgang konfiguriert werden
- Mit der positiven Taktflanke des Taktsignals an PIN1 gelangt das Ergebnis der Verknüpfung an den Ausgang
- Das Ausgangssignal aller Ausgänge wird zurückgekoppelt
- Ein Ausgang kann wahlweise aktiv High oder aktiv Low definiert werden
- Zur Bildung eines Produkttermes können alle Eingänge und alle Ausgangssignale verwendet werden
- Register-Ausgang : 8 Produktterme werden ODER -verknüpft
- Tristate-Ausgang : 7 Produktterme werden ODER -verknüpft
- Eine 0 am OE-Eingang gibt das Ergebnis der Register-Ausgänge frei

Laboraufgabe

Bearbeiten Sie folgende Aufgabenstellungen. Nutzen Sie dazu das GAL Development System und testen Sie den jeweils für die Aufgabenstellung programmierten GAL auf dem Experimentierboard.

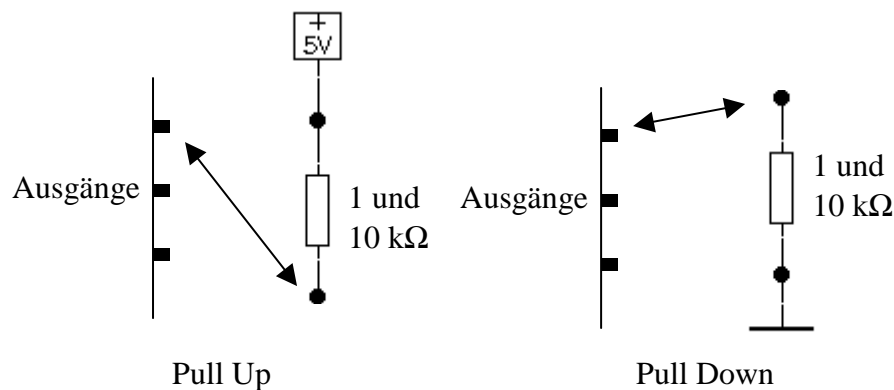
A1. AND mit 2 Eingängen, sowie ein AND mit 4 Eingängen wovon 2 Eingänge invertiert sind
Pinbelegung:

Eingang	E1 – Pin 2	Augang	Q1 – Pin 19
	E2 – Pin 3		
	/E3 – Pin 4		
	/E4 – Pin 5		

- A2. AND mit Tristate-Ausgang, beschreiben Sie verbal die Funktionsweise eines Tristate-Ausgangs und dessen Verwendung. Setzen Sie den Tristatezustand durch ein separates Eingangssignal. Messen Sie mit dem Oszilloskop, das Ausgangssignal des GALs. Dazu messen Sie einmal den Ausgang mit Pull Down Widerstand, einmal mit einem Pull Up Widerstand und einmal nur mit dem Messkopf des Oszilloskops. Stellen Sie die aufgenommenen Werte in einer Zeichnung dar und erläutern Sie das Verhalten.

Pinbelegung:

Eingang	E1 – Pin 2	Ausgang	Q1 – Pin 19
	E2 – Pin 3		
Tristate	T – Pin 4		



Bauen Sie dafür die Schaltung auf dem Experimentierbrett auf.

- A3 4 zu 1 Multiplexer mit Tristate-Ausgang, angelehnt an dem Baustein 74HCT151. Zusätzlich, stellen Sie an einem weiteren Ausgang das Signal des adressierten Eingangs invertiert dar. Beschreiben sie die Aufgabe eines Multiplexers. Welche praxisorientierte Anwendung könnte man damit realisieren.

Pinbelegung:

Dateneingang	E1 – Pin 2	Ausgang	Q1 – Pin 19
	E2 – Pin 3		/Q2 – Pin 18
	E3 – Pin 4		Q3 – Pin 17 Q1 ohne Tristate
	E4 – Pin 5		/Q4 – Pin 16 Q2 ohne Tristate
Adresseingang	A1 – Pin 6		
	A2 – Pin 7		
Tristate	T – Pin 8		

Bauen Sie die Schaltung auf dem Experimentierboard auf und testen Sie ihr Programm.

- A4 Machen Sie sich mit dem Datenblatt des 74HCT138 vertraut und realisieren sie dessen Funktionalität auf Basis eines GALs, beschreiben Sie die Funktionsweise Ihres GAL Programms und die Arbeitsweise des 74HCT138.

Pinbelegung:

Dateneingang	E1 – Pin 2	Ausgang	Q1 – Pin 19
	E2 – Pin 3		Q2 – Pin 18
	E3 – Pin 4		Q3 – Pin 17
Adresseingang	/A1 – Pin 5		Q4 – Pin 16

	/A2 – Pin 6		Q5 – Pin 15
	A3 – Pin 7		Q6 – Pin 14
			Q7 – Pin 13
			Q8 – Pin 12

Überprüfen Sie die Korrektheit Ihres Programms durch das Aufbauen und Testen der Schaltung auf dem Experimentierboard.

- A5 Implementierung eines Codewandler Dezimal zu BCD (8 Bit Eingang und 3 Bit Ausgang siehe Daten Blatt des 74HCT148)

Pinbelegung:

Dateneingang	E0 – Pin 1	Ausgang	Q1 – Pin 13
	E1 – Pin 2		Q2 – Pin 14
	E2 – Pin 3		Q3 – Pin 15
	E3 – Pin 4	Freigabeausgang	/EO – Pin 16
	E4 – Pin 5	Gruppensignalausgang	/GS – Pin 17
	E5 – Pin 6		
	E6 – Pin 7		
	E7 – Pin 8		
Freigabe der Eingänge	/EI – Pin 9		

Testen Sie auch diese Schaltung auf dem Experimentierboard und beschreiben Sie Ihren Weg der Umsetzung des Problems, welches Ihnen gestellt wurde.

- A6 Umsetzung einer 1Bit ALU (1Bit Halbaddierer, 1Bit Volladdierer, 1 Bit Halbsubtrahierer und 1 Bit Vollsubtrahierer).

Das Rechenwerk soll alle 4 Operationen beinhalten, wobei die Funktion über 2 Steuerleitungen gesteuert werden soll, zusätzlich benutzen die Rechenwerke für alle Operationen die gleichen Ein- und Ausgänge.

Erstellen Sie eine Wertetabelle für die minimale ALU und überlegen Sie sich wie viele UND und ODER Verknüpfungen Sie in Ihrem Programm benötigen.

Zur Umsetzung in das Programm bedenken Sie, das der GAL nur 8 ODER Verknüpfungen pro Ausgangszelle ermöglicht, benutzen Sie deshalb ein oder zwei zusätzliche kombinatorische Ein/Ausgangszellen.

Pinbelegung:

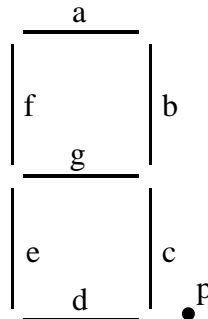
Dateneingang	E1 – Pin 2	Ausgang	Q1 – Pin 19
	E2 – Pin 3		Q2 – Pin 18
	E3 – Pin 4		Übertrag der nächste Stufe
	Übertrag der vorherigen Stufe		Temp1 – Pin 16
			Temp2 – Pin 15
Steuereingang	A1 – Pin 5		
	A2 – Pin 6		

		Funktion
/A1	/A2	Vollsubtrahierer
/A1	A2	Halbsubtrahierer
A1	/A2	Volladdierer
A1	A2	Halbaddierer

Testen Sie die Schaltung auf dem Experimentierbrett und im Simulator.

A7. Erstellen eines Siebensegmentdecoders (8-4-2-1 Code \Rightarrow 7 Segment Code)

Schreiben Sie die Umsetzung der Codes in einer Tabelle auf. Stellen Sie die Zahlen 0 bis 9 und die Buchstaben A bis F dar, benutzen Sie zur Unterscheidung ein kleines b und ein kleines d. Definieren Sie zusätzlich einen Testeingang, der alle LEDs zum Leuchten bringt und zusätzlich noch den Punkt.



Dateneingang	E1 – Pin 2	Ausgang	Q1 – Pin 19	Seg. p
	E2 – Pin 3		Q2 – Pin 18	Seg. g
	E3 – Pin 4		Q3 – Pin 17	Seg. f
	E4 – Pin 5		Q4 – Pin 16	Seg. e
Testeingang	T – Pin 6		Q5 – Pin 15	Seg. d
			Q6 – Pin 14	Seg. c
			Q7 – Pin 13	Seg. b
			Q8 – Pin 12	Seg. a

Testen Sie Ihr Programm auf der LED Leiterplatte.

A8. Implementierung eines RS-FlipFlops und eines getakteten RS-FlipFlops (Baustein 74L71) nur in unserem Versuch besitzt der Baustein nur einen Setz- bzw. Rücksetzeingang.

Erstellen Sie eine Wertetabelle für die beiden FlipFlops und setzen Sie diese dann in ein Programm um. Benutzen Sie zum Testen der Schaltung das Experimentierbrett und ein zusätzliches Taktsignal vom Funktionsgenerator.

Pinbelegung:

Dateneingang	E1 – Pin 2 Setzen	Ausgang	Q1 – Pin 19 RS FF
	E2 – Pin 3 Rücksetzen		Q2 – Pin 18 getaktetes RS FF
Takteingang	E3 – Pin 4		Temp1 – Pin 17
			Temp2 – Pin 16

A9. Überlegen Sie sich die Umsetzung eines Registerspeichers (TTL Baustein 74HCT574), welcher durch ein D-FlipFlop realisiert wird, überlegen Sie sich eine Realisierung des D-FlipFlops mittels RS-FlipFlops und zeichnen Sie die Schaltung auf.

Simulieren Sie den Registerspeicher des GALs mittels des eingebauten D-FlipFlops.

Pinbelegung:

Dateneingang	E1 – Pin 2	Ausgang	Q1 – Pin 19
Takteingang	CLK – Pin 1 Takt		
	/OE – Pin 11 Output enable		

Dokumentieren Sie die Ergebnisse der 3. Laboreinheit anhand der *.gal Dateien für die im einzelnen bearbeiteten Schaltungen und mittels Kurven des Oszilloskops.