

1. Trip-Genie

Der Besitzer einer kleinen Firma, die einige Außenmitarbeiter (Vertreter) beschäftigt, die insgesamt die vier Städte Hooterville, Mayberry, Mt. Pilot und Siler City bereisen müssen, hat folgendes Problem.

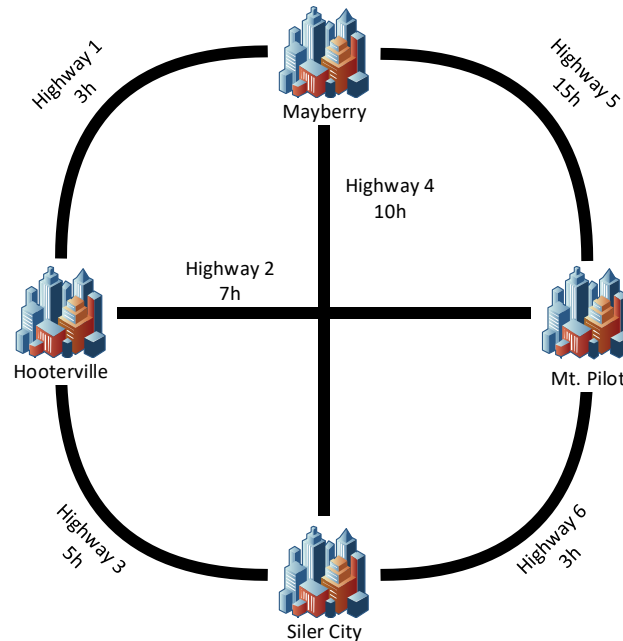


Abbildung 1: Die vier Städte und deren Verbindungshighways

Seine Vertreter wählen oft (zu oft) eine schlechte Route von Stadt zu Stadt aus und verbringen deswegen mehr Zeit auf der Straße als bei den Kunden. Er hat deswegen extra eine Karte angefertigt, die die Verbindungswege zwischen den Städten und die dafür nötigen Zeiten aufzeigt, vgl. Abbildung 1. Er möchte

nun ein kleines, tragbares Gerät, das jeder seiner Vertreter mitnehmen kann und ihm die schnellste

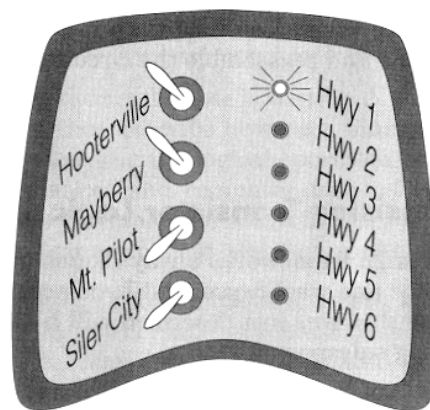


Abbildung 2: Grafischer Entwurf des TripGenie

Verbindung zwischen den Städten anzeigt. Dafür wurde sogar der grafische Entwurf aus Abbildung 2 angefertigt, der die Benutzungsoberfläche des Trip-Genie zeigt. Das Trip-Genie hat vier Schalter (einen für jede Stadt) und sechs LEDs (eine für jeden Highway). Der Vertreter schaltet zwei der vier Stadtschalter, um Start und Ziel seiner Reise anzuzeigen. Daraufhin aktiviert das Trip-Genie genau die LEDs für die Highways, welche auf der kürzesten Verbindung zwischen den beiden Städten liegen. Ihre Aufgabe besteht nun darin, die interne Schaltung des TripGenie zu realisieren.

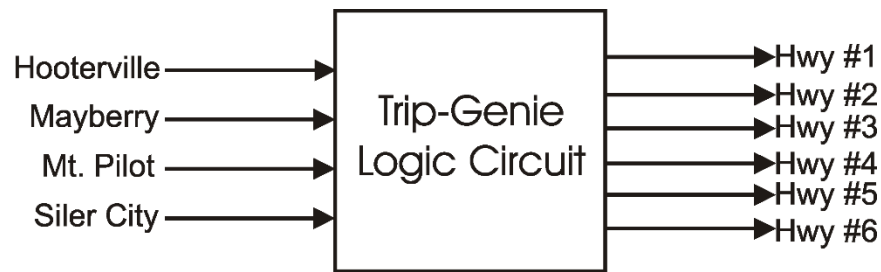


Abbildung 3: Eine High-Level-Darstellung des Trip-Genie mit seinen vier binären Ein- und sechs binären Ausgängen.

Aufgaben

- Entwerfen Sie ein Schaltnetz für die Realisierung des TripGenies in Form einer Wahrheitstafel. Als Eingaben dienen die Städte (H – Hootervillw, M – Mayberry, P – Mt. Pilot, S – Siler City) und als Ausgaben die einzelnen Highways (H_1, \dots, H_6), was auch der High-Level-Darstellung aus Abbildung 3 entnommen werden kann.
- Leiten Sie die einzelnen Schaltfunktionen der Highways als disjunktive Minimalformen her.
- Leider stehen Ihnen nur vier 74LS08 (4xUND; entspricht insgesamt 16 UND) Bausteine sowie ein 74LS32 (4xODER) Baustein zur Verfügung. Formen Sie Ihre Schaltfunktionen unter Ausnutzung des Distributiv-Gesetzes und der Verwendung von Koppeltermen (Terme, die in mehreren Schaltfunktionen vorkommen) so lange um, bis eine Realisierung des Schaltnetzes mit der gegebenen Hardware möglich ist.
- Simulieren Sie die umgeformte Variante des TripGenie aus Teilaufgabe c) mit Hilfe von [LogicCircuits](#).
- Entwerfen Sie einen Verdrahtungsplan für das Schaltnetz, indem Sie Abbildung 4 entsprechend ergänzen. Gehen Sie davon aus, dass Ihnen die Eingänge auch invertiert zur Verfügung stehen. **Testen Sie Ihren Verdrahtungsplan in CircuitJS. Nutzen Sie dafür das vorgegebene Template.**
- Ergänzen Sie den im Moodle in der Aufgabe gegebenen VHDL-Code um ihr Schaltnetz. Verwenden Sie dabei die gegebenen Strukturen sowie die case-when Direktive.

a)

H	M	P	S	H ₁	H ₂	H ₃	H ₄	H ₅	H ₆
0	0	1	1	0	0	0	0	0	1
0	1	0	1	1	0	1	0	0	0
0	1	1	0	1	1	0	0	0	0
1	0	0	1	0	0	1	0	0	0
1	0	1	0	0	1	0	0	0	0
1	1	0	0	1	0	0	0	0	0

b)

$$H_1 = \bar{H}M\bar{P}S + \bar{H}M\bar{P}\bar{S} + H\bar{M}\bar{P}\bar{S}$$

$$= M(\bar{H}\bar{P}S + \bar{H}\bar{P}\bar{S} + H\bar{P}\bar{S})$$

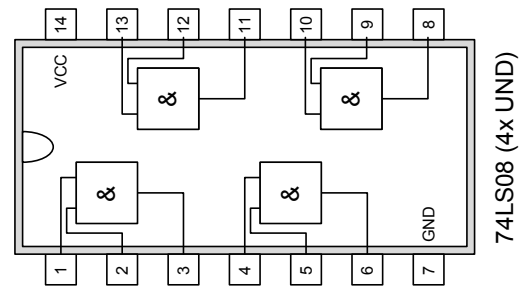
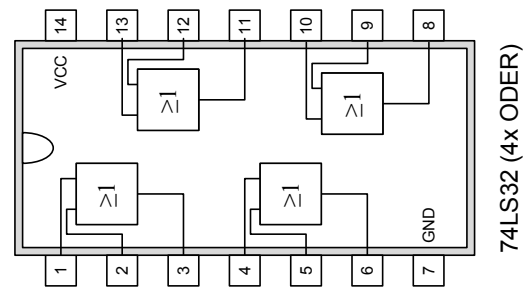
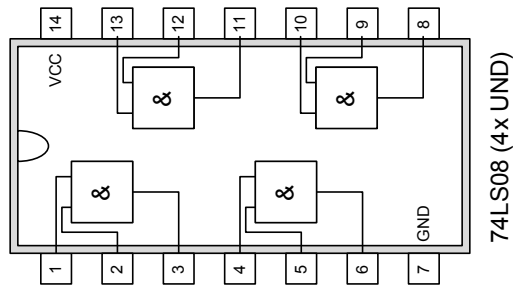
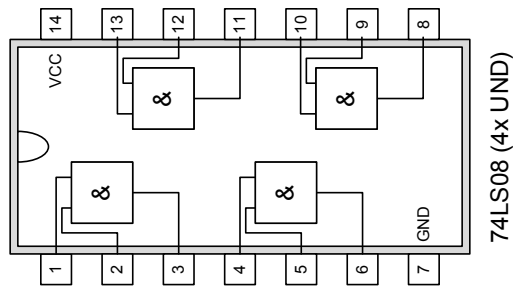
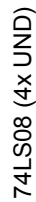
$$H_2 = \bar{H}M\bar{P}\bar{S} + H\bar{M}\bar{P}\bar{S} = \bar{P}\bar{S}(\bar{H}M + H\bar{M})$$

$$H_3 = \bar{H}M\bar{P}S + H\bar{M}\bar{P}S = \bar{P}S(\bar{H}M + H\bar{M})$$

$$H_4 = 0$$

$$H_5 = 0$$

$$H_6 = \bar{H}\bar{M}P\bar{S}$$



GND

Institut für Technische Informatik