

Sammanfattning

Föreläsning 10 - Digitalteknik

I boken: avsnitt 6.2 (kap 5.1 i Hemert)

Tillståndskodning

Syftet med föreläsningen är att förstå att olika sätt att koda tillstånden i en graf (ge dem binära värden) ger olika stora realiseringar. Vi lär oss några metoder som ofta ger bra tillståndskodningar (alltså en liten realisering).

En NBCD kodning ger bara tillstånden binära värden efter hur vi numrerat dem $s_0 = 0, s_1 = 1, \dots$, där antalet bitar n som används beror på det totala antalet tillstånd enligt $n = \lceil \log_2 |S| \rceil$, där $|S|$ är antalet tillstånd. Exempelvis 3 tillstånd med $s_0 = 00, s_1 = 01$ och $s_2 = 10$.

Observation: Antalet implikatorer i en realisering av en maskin beror på tillståndskodningen.

Det finns inga kända metoder för att ta fram en optimal tillståndskodning.

Idé för att hitta en bra men inte nödvändigtvis optimal tillståndskodning: Hitta en kodning där så få variabler som möjligt ändras vid tillståndsövergångar.

Metod Gray-kodning: Gör en uppräkningsordning av tillstånden så att tillstånd med övergångar ligger efter varandra i uppräkningsordningen (så gott det går). Koda sedan tillstånden med en Gray-kod.

Metod Reduced Dependency: I denna metod försöker vi hitta beroenden mellan tillstånden som sedan kan ligga till grund för tillståndskodningen.

RD-Algoritmen För varje par av tillstånd gör vi följande

- anta att de två tillstånden är i samma block.
- iterativt, för varje insignalkombination, gruppera nästa tillstånd hos ett block till ett nytt block. Om ett tillstånd finns i flera block måste alltså dessa block slås ihop till ett block.

De erhållna uppdelningarna svarar mot beroenden i grafen. Använd dem för att hitta en lämplig tillståndskodning.

1-Hot tillståndskodning: Använd en tillståndsvariabel för varje tillstånd. Koda sedan så att alla övergångar ändrar (högst) två variabler. Alltså, $s_0 = 10 \dots 0, s_1 = 010 \dots 0$, etc. Använder fler D-element än nödvändigt och är i allmänhet inte så bra.