

EDA322 Digital konstruktion

Några uppgifter om tillståndsmaskiner

Uppgifterna är hämtade ur Brown, Vranesic: Fundamentals of Digital Logic with VHDL Design ed 3, kapitel 8.

- 1 Ta fram en krets uppbyggd av D-vippor som implementerar tillståndstabellen i Figur 1. Skriv VHDL-kod för implementeringen

Present state		Next state				Output
		w=0		w=1		
y ₂	y ₁	y ₂	y ₁	y ₂	y ₁	z
0	0	1	0	1	1	0
0	1	0	1	0	1	0
1	0	1	1	0	0	0
1	1	1	0	0	1	1

Figur 1 Tillståndstabell

-
- 2 Upprepa uppgift 1 med hjälp av JK-vippor
-

- 3 Upprepa uppgift 1 med hjälp av T-vippor
-

- 4 Lös uppgift 1 med beteendemässig VHDL-kod
-

- 5 Ta fram en tillståndsdigrammet för en tillståndsmaskin av Mooretyp som ger utsignalen z värdet ett (1) om vi via ingången w har detekterat någon av sekvenserna 110 eller 101. Sekvenserna kan vara överlappande
-

- 6 Upprepa uppgift 5 med hjälp av en tillståndsmaskin av Mealytyp
-



- 7 Minimera tillståndstabellen i Figur E7 med de två insignalerna *D* och *N* och utsignalen *z*

Present state	Next state				Output
	DN				
	00	01	10	11	
S1	S1	S3	S2	-	0
S2	S2	S4	S5	-	0
S3	S3	S6	S7	-	0
S4	S1	-	-	-	1
S5	S3	-	-	-	1
S6	S6	S8	S9		0
S7	S1	-	-	-	1
S8	S1	-	-	-	1
S9	S3	-	-	-	1

Figur E7

- 8 Skriv VHDL-kod för både den ominimerade och den minimerade tillståndstabellen i uppgift 7

- 9 Ta fram tillståndstabellen för en tillståndsmaskin som skall detektera en sekvens av fyra värden på ingången *w*. Utsignalen *z* ska bli ett (1) om sekvensen är 0010 eller 1110. I övriga fall ska utsignalen bli noll (0). Efter en sekvens om fyra värden ska tillståndsmaskinen börja om med en ny sekvens, dvs sekvenserna kan inte vara överlappande. Minimera tillståndstabellen

- 10 Ta fram och minimera tillståndstabellen för en tillståndsmaskin som detekterar om antalet ettor (1) i en trebitars sekvens på ingången *w* är udda eller jämnt. Udda antal ettor skall sätta utsignalen *z* till ett (1). Trebitarssekvenserna kan inte vara överlappande

- 11 Minimera tillståndstabellen i figur E11

Present state	Next state		Output	
	$w_1=0$	$w_1=1$	$w_1=0$	$w_1=1$
A	B	C	0	0
B	D	-	0	0
C	F	E	0	1
D	B	G	0	0
E	F	C	0	1
F	E	D	0	1
G	F	-	0	1

Figur E11 Modifierad tillståndstabell

12 Använd D-vippor för att konstruera en modulo-6-räknare med sekvensen 0, 1, 2, 3, 4, 5, 0, 1,....
Räknaren räknar klockpulser men bara då ingången w är ett (1).

13 Upprepa *uppgift 12* med JK-vippor

14 Upprepa *uppgift 12* med T-vippor

15 Ta fram tillståndsgraf och tillståndstabell för en räknarliknande krets med åtta (8) tillstånd 0-7. Är ingången w ett (1) så skall kretsen räkna uppåt med steget två (2). Då den har mått maxvärde så skall den göra wrap around, dvs är värdet åtta (8) så skall nästa värde bli noll (0) medan nästa värde skall bli ett (1) om nuvarande värde är nio (9). Är w noll (0) så skall kretsen fungera som en vanlig nedräknare med steget ett som gör wrap around då den når noll (0).
Implementera kretsen med hjälp av D-vippor

16 Skriv VHDL-kod för tillståndstabellen framtagna i *uppgift 15*