

Tentamen Digitalkonstruktion I 3p

Datum: 2000-08-26

Tid: 5 timmar

Hjälpmedel: Miniräknare

Kursansvarig: Bengt Oelmann, tel: 148792 eller 171311, e-post: bengt@ite.mh.se

max. antal poäng: 20 p

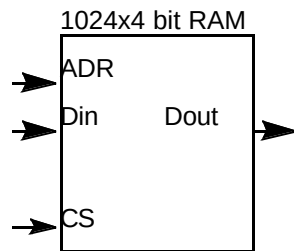
Anvisningar för inlämnade lösningar:

- Resonemang och motiveringar får ej vara så knapphändiga att de blir svåra att följa.
- Införda beteckningar skall definieras.
- Tankegången bakom uppställda ekvationer skall förklaras.
- Uträkningarna skall vara tillräckligt fullständiga för att visa hur slutresultatet erhållits.
- Approximationer ska motiveras och underkastas efterkontroll.
- Ange svaren med lämpligt antal gällande siffror
- Varje problemlösning skall avslutas med ett klart formulerat svar.

UPPGIFTER

1. Skriv VHDL-kod för en kombinatorisk krets som omvandlar tal mellan 0 och 9 givna i Excess-3-kod till BCD-kod. Se bifogad tabell. (4 p)

2. Konstruera minnesavkodningen för ett ROM minne med storleken 4096x8 bitar (4kbytes) baserat på 1024x4 bitar RAM komponenter (1kbyte). Rita blockdiagram. (4p)



ADR: adress, 10 bitar

Dout: data ut, 4 bitar

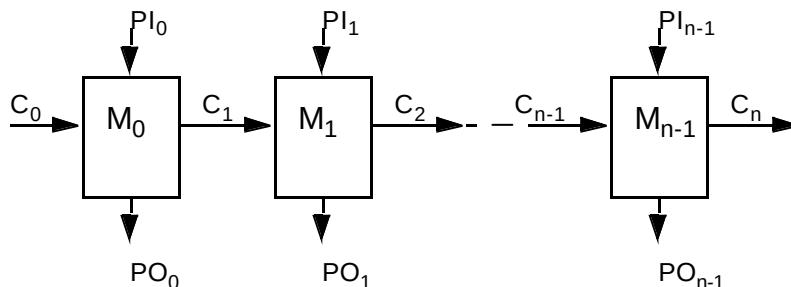
CS: *Chip Select*, CS=1: aktiverar minneskretsen, CS=0: Dout är i three-state (Z).

3. Konstruera en räknare som kan både räkna i excess-3 kod (se bifogad tabell) och binär kod. Med en kontrollsignal som vi kallar R styrs räknaren. Om $R=0$ ska den räkna i binär kod och om $R=1$ ska den räkna i excess-3 kod. Räknaren ha följande räknesekvens ... 0, 1, ... 8, 9, 0, Använd 74x163 (se bifogad beskrivning) samt lämpliga grindar för att bygga räknaren. Rita ett schema för din konstruktion. (4p)

4. Vad är det maximalt tillåtna *clock-skew* i ett skiftregister uppbyggd av D-vippor med följande data: (2p)

fördröjning	[ns]	kommentar
t_{hold}	2	hålltid
t_{setup}	1	setup-tid
t_{pCQ}	5	fördröjning från klockflank till Q-utgång

5. En iterativ krets är en kombinatorisk krets som har strukturen som visas i figuren nedan. PI är primära ingångar, PO är primära utgångar och C är kaskadkopplingssignaler mellan modulerna M_i .



Den här strukturen är lämplig för problem som kan lösas med en enkel algoritm formulerad som:

- (1) Sätt C_0 till dess initialvärde och sätt $i=0$.
- (2) Använd C_i och PI_i för att bestämma värdena på PO_i och C_{i+1} .
- (3) öka i med 1
- (4) Om $i < n$ gå till steg 2, annars avsluta.

En n-bitars komparator är lämplig att göra som en iterativ krets.

- a) Beskriv en n-bitars komparator m.h.a en algoritm på samma format som den generella given ovan. (2 p)
- b) Ta fram en lösning på grindnivå för modulen M_i som ska användas i n-bitars komparatorn. Rita ett blockschema för den iterativa komparatorlösningen (enligt den generella strukturen given tidigare i uppgiften) för en fyra bitars komparator samt ett schema på grindnivå för M_i . (2 p)
- c) Forma om den iterativa lösningen till en sekventiell lösning. Redovisas i form av VHDL kod. (2 p)

Appendix:

Decimal digit	BCD (8421)	2421	Excess-3	Biquinary	1-out-of-10
0	0000	0000	0011	0100001	1000000000
1	0001	0001	0100	0100010	0100000000
2	0010	0010	0101	0100100	0010000000
3	0011	0011	0110	0101000	0001000000
4	0100	0100	0111	0110000	0000100000
5	0101	1011	1000	1000001	0000010000
6	0110	1100	1001	1000010	0000001000
7	0111	1101	1010	1000100	0000000100
8	1000	1110	1011	1001000	0000000010
9	1001	1111	1100	1010000	0000000001

Table 9-1 State table for a 74x163 4-bit binary counter.

Inputs				Current State				Next State			
/CLR	/LD	ENT	ENP	QD	QC	QB	QA	QD*	QC*	QB*	QA*
0	x	x	x	x	x	x	x	0	0	0	0
1	0	x	x	x	x	x	x	D	C	B	A
1	1	0	x	x	x	x	x	QD	QC	QB	QA
1	1	x	0	x	x	x	x	QD	QC	QB	QA
1	1	1	1	0	0	0	0	0	0	0	1
1	1	1	1	0	0	0	1	0	0	1	0
1	1	1	1	0	0	1	0	0	0	1	1
1	1	1	1	0	0	1	1	0	1	0	0
1	1	1	1	0	1	0	0	0	1	0	1
1	1	1	1	0	1	0	1	0	1	1	0
1	1	1	1	0	1	1	0	0	1	1	1
1	1	1	1	0	1	1	1	1	0	0	0
1	1	1	1	1	0	0	0	1	0	0	1
1	1	1	1	1	0	0	1	1	0	1	0
1	1	1	1	1	0	1	0	1	0	1	1
1	1	1	1	1	0	1	1	1	1	0	0
1	1	1	1	1	1	0	0	1	1	0	1
1	1	1	1	1	1	0	1	1	1	1	0
1	1	1	1	1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1	0	0	0	0

The RCO (Ripple carry out") signal indicates a carry from the most significant bit position, and is 1 when all of the count bits are 1 and ENT is 1.

