Rechnerstrukturen WS 2011/12

- Boolesche Funktionen und Schaltnetze
- KV-Diagramme (Wiederholung)
- Algorithmus von Quine und McCluskey
  - Einleitung, Berechnung von PI
  - PI-Tafel für das Überdeckungsproblem
- Unvollständig definierte Funktionen
  - Einleitung
  - Minimalpolynome für partiell definierte Funktionen
- Hazards
  - Einleitung
  - Schaltungshazards

Hinweis: *Folien teilweise a. d. Basis von Materialien von Thomas Jansen*

29. August 2011

Minimalpolynombestimmung mit KV-Diagramm

Wieso suchen wir eigentlich Minimalpolynome?

Aufgabe Bestimme für $f\colon \{0,1\}^4 \to \{0,1\}$ ein Minimalpolynom.

Vorgehen

1. Eintragen der Funktion ins KV-Diagramm
2. Finden aller maximaler Zweierpotenz-Rechtecke
3. Finden eines Primimplikanten für jedes Rechteck
4. Finden einer Überdeckung aller Einsen durch eine minimale Monomauswahl

jetzt noch ein Beispiel
$f\colon \{0,1\}^4 \to \{0,1\}$ mit $(1,0,1,1,0,1,0,1,0,1,0,1,0,0,1,1)$

Eintragen der Funktion ins KV-Diagramm

$$f: \{0,1\}^4 \to \{0,1\} \text{ mit } \begin{array}{c} \begin{array}{cccccccccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \end{array} \\ (1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1) \end{array}$$

$x_3\,x_4$ \ $x_1\,x_2$	00	01	11	10
00	1			
01		1		1
11	1	1	1	1
10	1		1	

$0 = (00\,00)_2$
$1 = (00\,01)_2$
$2 = (00\,10)_2$
$3 = (00\,11)_2$
$4 = (01\,00)_2$
$5 = (01\,01)_2$
$6 = (01\,10)_2$
$7 = (01\,11)_2$
$8 = (10\,00)_2$
$9 = (10\,01)_2$
$10 = (10\,10)_2$
$11 = (10\,11)_2$
$12 = (11\,00)_2$
$13 = (11\,01)_2$
$14 = (11\,10)_2$
$15 = (11\,11)_2$

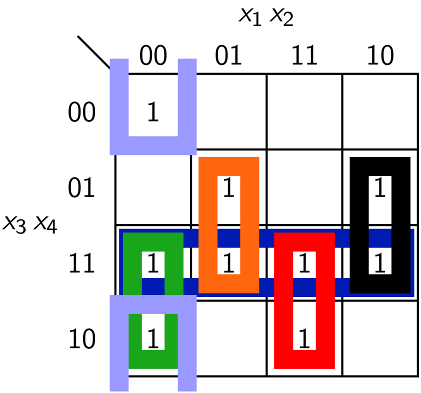
Finden aller maximaler Zweierpotenz-Rechtecke

$f: \{0,1\}^4 \to \{0,1\}$ mit $(1,0,1,1,0,1,0,1,0,1,0,1,0,0,1,1)$

$x_3\,x_4$ \ $x_1\,x_2$	00	01	11	10
00	1			
01		1		1
11	1	1	1	1
10	1		1	

Finden eines Primimplikanten für jedes Rechteck

$f: \{0,1\}^4 \to \{0,1\}$ mit $(1,0,1,1,0,1,0,1,0,1,0,1,0,0,1,1)$



$x_3\,x_4$

$\overline{x_1}\,\overline{x_2}\,x_3$

$\overline{x_1}\,x_2\,x_4$

$x_1\,x_2\,x_3$

$x_1\,\overline{x_2}\,x_4$

$\overline{x_1}\,\overline{x_2}\,\overline{x_4}$

Überdeckung aller Einsen durch minimale Monomauswahl

$f\colon \{0,1\}^4 \to \{0,1\}$ mit $(1,0,1,1,0,1,0,1,0,1,0,1,0,0,1,1)$

$x_3\,x_4$ \ $x_1\,x_2$	00	01	11	10
00	1			
01		1		1
11	1	1	1	1
10	1		1	

$x_3\,x_4$	beste Wahl
$\overline{x_1}\,\overline{x_2}\,x_3$	
$\overline{x_1}\,x_2\,x_4$	**erforderlich**
$x_1\,x_2\,x_3$	**erforderlich**
$x_1\,\overline{x_2}\,x_4$	**erforderlich**
$\overline{x_1}\,\overline{x_2}\,\overline{x_4}$	**erforderlich**

also $x_3\,x_4 \vee \overline{x_1}\,x_2\,x_4 \vee x_1\,x_2\,x_3 \vee x_1\,\overline{x_2}\,x_4 \vee \overline{x_1}\,\overline{x_2}\,\overline{x_4}$
Minimalpolynom

Fazit KV-Diagramme

Mit KV-Diagrammen effizient beide Schritte zur Minimalpolynomberechnung durchführbar

1. Bestimmung aller Primimplikanten
2. Bestimmung einer minimalen Überdeckung

für Funktionen $f\colon \{0,1\}^n \to \{0,1\}$ für $n \in \{3,4\}$

klar Das reicht nicht aus.

Wie bestimmen wir Minimalpolynome für $f\colon \{0,1\}^n \to \{0,1\}$ für größeres n?

klar Wir suchen einen Algorithmus.

Der Algorithmus von Quine und McCluskey

Willard van Orman Quine (1955)
Edward J. McCluskey (1956)

Algorithmus von Quine/McCluskey

Eingabe Fkt. f als Liste aller Minterme zu einschlägigen Indizes
Ausgabe Minimalpolynom zu f

1. Berechne PI, Menge aller Primimplikanten von f.
2. Berechne minimale f überdeckende Auswahl aus PI.

Algorithmus von Quine/McCluskey: Erster Teil

Algorithmus 11 (Berechnung von PI)

Eingabe L_0: Liste aller Minterme zu einschlägigen Indizes von f
Ausgabe PI: Menge aller Primimplikanten zu f

1. $i := 0$; PI $:= \emptyset$
2. So lange $L_i \neq \emptyset$
3. $\quad L_{i+1} := \{m \mid \exists x \colon \{m\,x, m\,\overline{x}\} \subseteq L_i\}$ (Resolution)
4. $\quad \mathrm{PI} := \mathrm{PI} \cup \{m \in L_i \mid m$ hat keine echte Verkürzung in $L_{i+1}\}$
5. $\quad i := i + 1$
6. Ausgabe PI

Fehlt: Beweis, dass Algorithmus 11 korrekt ist.

Korrektheit der PI-Berechnung

1. $i := 0$; $\text{PI} := \emptyset$
2. So lange $L_i \neq \emptyset$
3. $\quad L_{i+1} := \{m \mid \exists x \colon \{m\,x, m\,\overline{x}\} \subseteq L_i\}$
4. $\quad \text{PI} := \text{PI} \cup \{m \in L_i \mid m \text{ hat keine echte Verkürzung in } L_{i+1}\}$

...

Behauptung L_i ist Menge aller Implikanten der Länge $n - i$

Induktionsanfang stimmt für L_0 ✓

Induktionsschluss Annahme stimmt für L_i

klar L_{i+1} enthält nur Monome der Länge $n - i - 1 = n - (i + 1)$

Beobachtung L_{i+1} enthält nur Implikanten
denn $m\,x$ Implikant und $m\,\overline{x}$ Implikant
$\Rightarrow m$ Implikant (Resolution)

Beobachtung L_{i+1} enthält alle Implikanten
denn jeder Implikant m hat Verlängerung um 1 in L_i,
(Induktionsvoraussetzung), also kommt m nach L_{i+1} ✓

Korrektheit der PI-Berechnung

1. $i := 0$; $\text{PI} := \emptyset$
2. So lange $L_i \neq \emptyset$
3. $\quad L_{i+1} := \{m \mid \exists x\colon \{m\,x, m\,\overline{x}\} \subseteq L_i\}$
4. $\quad \text{PI} := \text{PI} \cup \{m \in L_i \mid m \text{ hat keine echte Verkürzung in } L_{i+1}\}$

...

Wir haben $\quad L_i$ ist Liste Implikanten der Länge $n - i$

klar $\quad$ Algorithmus 11 terminiert
denn spätestens $L_{n+1} = \emptyset$

Beobachtung

am Ende enthält PI alle Primimplikanten
denn jeder Primimplikant ist Implikant, also in einer Liste vorhanden
und Primimplikant hat keine Verkürzung, die auch Implikant ist,
wird deshalb zu PI hinzugefügt

also $\quad$ Berechnung der PI nach Quine/McCluskey korrekt $\checkmark$

Beispiel PI-Berechnung

1. $i := 0$; $\mathrm{PI} := \emptyset$
2. So lange $L_i \neq \emptyset$
3. $\quad L_{i+1} := \{m \mid \exists x\colon \{m\,x, m\,\overline{x}\} \subseteq L_i\}$
4. $\quad \mathrm{PI} := \mathrm{PI} \cup \{m \in L_i \mid m$ hat keine echte Verkürzung in $L_{i+1}\}$
5. $\quad i := i + 1$
6. Ausgabe PI

$$\mathrm{PI} = \{\overline{x_1}\,\overline{x_2}\,x_3, \overline{x_1}\,x_2\,x_4, x_1\,\overline{x_2}\,x_4, x_1\,x_2\,x_3, x_3\,x_4\}$$

$$\begin{aligned}
L_0 = \{&\overline{x_1}\,\overline{x_2}\,x_3\,\overline{x_4}, \overline{x_1}\,\overline{x_2}\,x_3\,x_4, \overline{x_1}\,x_2\,\overline{x_3}\,x_4, \overline{x_1}\,x_2\,x_3\,x_4,\\
&x_1\,\overline{x_2}\,\overline{x_3}\,x_4, x_1\,\overline{x_2}\,x_3\,x_4, x_1\,x_2\,x_3\,\overline{x_4}, x_1\,x_2\,x_3\,x_4\}\\
L_1 = \{&,\overline{x_1}\,\overline{x_2}\,x_3, \overline{x_1}\,x_3\,x_4, \overline{x_2}\,x_3\,x_4, \overline{x_1}\,x_2\,x_4, x_2\,x_3\,x_4, x_1\,\overline{x_2}\,x_4,\\
&x_1\,x_3\,x_4, x_1\,x_2\,x_3\}\\
L_2 = \{&x_3\,x_4\}\\
L_3 = \emptyset&
\end{aligned}$$

Der Algorithmus von Quine und McCluskey

Algorithmus von Quine/McCluskey

Eingabe Fkt. f als Liste aller Minterme zu einschlägigen Indizes
Ausgabe Minimalpolynom zu f

1. Berechne PI, Menge aller Primimplikanten von f. $\checkmark$
2. Berechne minimale f überdeckende Auswahl aus PI.

schwieriges kombinatorisches Problem

nur heuristische Lösung
mit gewissen Freiheitsgraden

Das Überdeckungsproblem

Aufgabe Finde minimale Überdeckung von f mit PI.

PI-Tafel

- eine Zeile für jeden Primimplikanten
- eine Spalte für jede Eins-Eingabe
- Eintrag $= \begin{cases} 1 & \text{Primimplikant überdeckt Eins-Eingabe} \\ 0 & \text{sonst} \end{cases}$

klar PI-Tafel meist riesig groß

klar verkleinern

dazu Verkleinerungsregeln

Freiheit durch

- Reihenfolge der Regelanwendung
- Art der Regelanwendung

Problem Lösung oft trotzdem nicht ablesbar

Lösung Backtracking oder Heuristiken

Verkleinerungsregeln

- Streichung von Kernimplikanten-Zeilen
  Spalte s mit nur einer 1: Wähle Primimplikant der korrespondierenden Zeile, streiche diese Zeile und alle von ihr überdeckten Spalten.
  Begründung Kernimplikanten müssen gewählt werden. Überdeckte Eins-Eingaben können gestrichen werden.
- Streichung überdeckender Spalten
  Spalten s, s' mit $s \geq s'$: Streiche Spalte s.
  Begründung s' ist schwieriger zu überdecken: Jeder Primimplikant, der s' überdeckt, überdeckt auch s. Also können wir s entfernen.
- Streichung überdeckter Zeilen
  Zeilen z, z' mit $z \geq z'$: Streiche Zeile z'.
  Begründung Primimplikant zu z deckt mehr ab als Primimplikant zu z'. Also ist es sicher nicht schlechter, z zu wählen.

Beispiel PI-Tafel

	0010	0011	0101	0111	1001	1011	1110	1111
$\overline{x_1}\,\overline{x_2}\,x_3$	1	1						
$x_3\,x_4$		1		1		1		1
$\overline{x_1}\,x_2\,x_4$			1	1				
$x_1\,x_2\,x_3$							1	1
$x_1\,\overline{x_2}\,x_4$					1	1		

$<$

Streichung überdeckender Spalten

Beispiel PI-Tafel

	0010	0101	0111	1001	1011	1110	1111
$\overline{x_1}\,\overline{x_2}\,x_3$	1						
$x_3\,x_4$			1		1		1
$\overline{x_1}\,x_2\,x_4$		1	1				
$x_1\,x_2\,x_3$						1	1
$x_1\,\overline{x_2}\,x_4$				1	1		

$<$

Streichung überdeckender Spalten

Beispiel PI-Tafel

	0010	0101	1001	1011	1110	1111
$\overline{x_1}\ \overline{x_2}\ x_3$	1					
$x_3\ x_4$				1		1
$\overline{x_1}\ x_2\ x_4$		1				
$x_1\ x_2\ x_3$					1	1
$x_1\ \overline{x_2}\ x_4$			1	1		

$<$

Streichung überdeckender Spalten

Beispiel PI-Tafel

	0010	0101	1001	1110	1111
$\overline{x_1}\,\overline{x_2}\,x_3$	1				
$x_3\,x_4$					1
$\overline{x_1}\,x_2\,x_4$		1			
$x_1\,x_2\,x_3$				1	1
$x_1\,\overline{x_2}\,x_4$			1		

Streichung überdeckter Zeilen

Beispiel PI-Tafel

	0010	0101	1001	1110	1111
$\overline{x_1}\,\overline{x_2}\,x_3$	1				
$\overline{x_1}\,x_2\,x_4$		1			
$x_1\,x_2\,x_3$				1	1
$x_1\,\overline{x_2}\,x_4$			1		

Kernimplikanten-Zeilen

also $f(x_1,\ldots,x_4) = \overline{x_1}\,\overline{x_2}\,x_3 \vee \overline{x_1}\,x_2\,x_4 \vee x_1\,x_2\,x_3 \vee x_1\,\overline{x_2}\,x_4$

Unvollständig definierte Funktionen

formal $f\colon \{0,1\}^n \to \{0,1,*\}$

Bedeutung $f(x) = *$ Funktionswert für x egal ("*don't care*")

klar Schaltnetz realisiert immer eine
„normale" boolesche Funktion

Definition

$f'\colon \{0,1\}^n \to \{0,1\}$ heißt Realisierung (oder auch Erweiterung) von
$f\colon \{0,1\}^n \to \{0,1,*\} \Leftrightarrow \forall x \in \{0,1\}^n\colon\ (f'(x) = f(x)) \vee (f(x) = *)$

klar größere Freiheiten

Wie nutzen wir die?

Wahl der Realisierung

Wunsch Finde Realisierung f' mit minimalem Minimalpolynom.

nützliche Begriffe

- minimale Erweiterung f_0 von f

$$f_0(x) := \begin{cases} f(x) & \text{falls } f(x) \in \{0,1\} \\ 0 & \text{sonst} \end{cases}$$

- maximale Erweiterung f_1 von f

$$f_1(x) := \begin{cases} f(x) & \text{falls } f(x) \in \{0,1\} \\ 1 & \text{sonst} \end{cases}$$

Über extreme Erweiterungen

Beobachtungen

- Primimplikanten von f_0 sind keine echten Verkürzungen von Primimplikanten von f_1
- Primimplikanten von f_1 können echte Verkürzungen von Primimplikanten von f_0 sein

Warum?

klar zusätzliche Einsen verkürzen Primimplikanten höchstens

Also hat f_1 das kürzere Minimalpolynom?

So allgemein gilt das nicht!

denn f_1 kann viel mehr Primimplikanten haben

Über unvollständige definierte Funktionen

Satz 12

Minimalpolynome einer unvollständig definierten Funktion
$f\colon \{0,1\}^n \to \{0,1,*\}$ enthalten ausschließlich Primimplikanten von f_1.

Beweis Sei $p = m_1 \vee m_2 \vee \cdots \vee m_k$ Minimalpolynom von f.

Beobachtung p ist Realisierung von f.

Sei m_i beliebiger Primimplikant in p.

Beobachtung m_i ist Implikant von f_1.

denn $m_i(x) = 1 \Rightarrow (f(x) = 1) \vee (f(x) = *) \Rightarrow f_1(x) = 1$

also Verkürzung m' von m_i ist Primimplikant von f_1

aber Ersetzen wir m_i in p durch m', erhalten wir Polynom für f

klar m' kann nicht echte Verkürzung von m_i sein,
sonst wäre p nicht Minimalpolynom zu f

also m_i Primimplikant von f_1 □

Minimalpolynome unvollständig definierter Funktionen

Algorithmus

1. Berechne Menge aller Primimplikanten $\mathrm{PI}(f_1)$ zu f_1.
2. Berechne minimale Überdeckung von f_0 durch Monome aus $\mathrm{PI}(f_1)$.

also algorithmisch keine neuen Probleme

Einwand Vielleicht lässt sich f_0 gar nicht so überdecken?

Beobachtung Überdeckung von Nullen von f_0 unproblematisch

Wieso eigentlich?

Primimplikanten von f_1 können nur Einsen oder *don't cares* von f überdecken!

Hazards

ärgerlich, aber unvermeidlich   Abweichungen zwischen Modell und Realität

Das gilt auch für technisch realisierte Schaltnetze.

Was passiert, wenn die Eingabe wechselt?

klar   Die Ausgabe kann wechseln.

Wunsch

- wenn f die Ausgabe wechselt, wechselt das Schaltnetz genau einmal seine Ausgabe
- wenn f die Ausgabe nicht wechselt, wechselt das Schaltnetz seine Ausgabe nicht

Realität   Schaltnetz kann Ausgabe „zwischendurch“ wechseln.

Das heißt Hazard.

Hazards: Begriffe

Hazards sind

- statisch   Ausgabewert soll gleich bleiben, ändert sich aber.
- dynamisch   Ausgabewert soll sich ändern, ändert sich aber mehrfach.
- Funktionshazard   „schon in der Funktionsdefinition enthalten“
- Schaltungshazard   Hazard, der kein Funktionshazard ist

also   vier verschiedene Fälle

Statischer Funktionshazard

Definition

$f : \{0,1\}^n \to \{0,1\}$ hat statischen Funktionshazard, wenn es $a = a_1 a_2 \cdots a_n \in \{0,1\}^n$ und $b = b_1 b_2 \cdots b_n \in \{0,1\}^n$ mit $a \neq b$ und $f(a) = f(b)$ gibt, sowie $c = c_1 c_2 \cdots c_n \in \{0,1\}^n$ mit $c_i \in \{a_i, b_i\}$ für alle $i \in \{1, 2, \ldots, n\}$ und $f(c) \neq f(a)$.

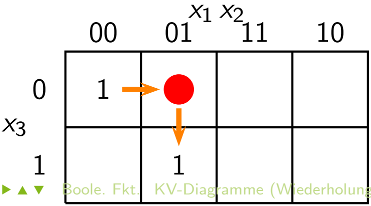
Beispiel

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

$a = 000,\ f(a) = 1$

$c = 010,\ f(c) = 0$

$b = 011,\ f(b) = 1$



Dynamischer Funktionshazard

Definition

$f\colon \{0,1\}^n \to \{0,1\}$ hat dynamischen Funktionshazard, wenn es $a, b \in \{0,1\}^n$ mit $a \neq b$ und $f(a) \neq f(b)$ gibt, sowie $c = c_1 c_2 \cdots c_n$ und $c' = c'_1 c'_2 \cdots c'_n$ mit $c_i \in \{a_i, b_i\}$, $c'_i \in \{c_i, b_i\}$ $(i \in \{1, 2, \ldots, n\})$ und $f(c) \neq f(a)$ sowie $f(c) \neq f(c')$.

Beispiel

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

$a = 000,\ f(a) = 1$

$c = 001,\ f(c) = 0$

$c' = 011,\ f(c') = 1$

$b = 111,\ f(b) = 0$

x_3 \ $x_1 x_2$	00	01	11	10
0	1		1	
1		1	0	

Zusammenfassung: Statische/Dynamische Hazards

Wir haben korrektes Schaltnetz S für f.

Wir betrachten Eingabewechsel von a auf b.

1. Funktionswert bleibt gleich: $f(a) = f(b)$
   Am Ausgang von S liegt kurzzeitig ein anderer Wert an.
   statischer Hazard
2. Funktionswert ändert sich: $f(a) \neq f(b)$
   Am Ausgang von S ändert sich der Wert mehrfach.
   dynamischer Hazard

Weitergehende Klassifikation

- Manche Hazard heißen Funktionshazards.
- Die übrigen Hazards heißen Schaltungshazards.

Wie kommt es zum Funktionshazard?

Betrachte Eingabewechsel von a nach b mit
entweder $f(a) = f(b)$ statisch
oder $f(a) \neq f(b)$ dynamisch

Es gibt statisch: 1 andere Eingabe c
dynamisch: 2 andere Eingaben c, c'

zwischen a und b,
so dass der Funktionswert beim Weg von a nach b

statisch über c wechselt
dynamisch über c und c' mehrfach wechselt

zentral andere Eingaben echt „dazwischen“

Was bedeutet „dazwischen“?

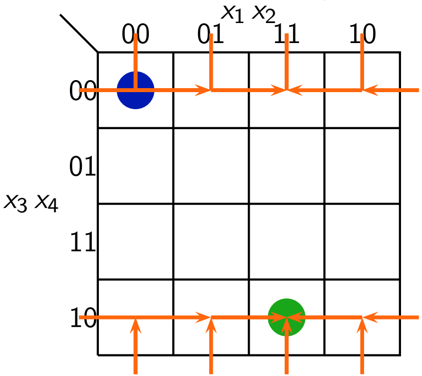
in Bezug auf Eingabewechsel

- Wertänderung einiger Eingabebits (alle Bits mit $a_i \neq b_i$)
- nicht völlig gleichzeitig
- also „dazwischen“ = kann als Zwischenschritt bei Wertänderung der Eingabebits vorkommen

beim Blick auf KV-Diagramm

- auf einem kürzsten Weg von a nach b

Kürzeste Wege im KV-Diagramm



$a = 0000$

$b = 1110$

Beobachtung: 3 Bits verschieden
kürzeste Wege haben Länge 3
(i.a.: n unterschiedliche Bits $\Rightarrow$ kürzeste Weglänge $= n$)

Funktionshazards

klar Funktionshazards sind in Realisierungen von Schaltnetzen nicht zu vermeiden

zur Kenntnis nehmen Es gibt Hazards.

für uns interessanter Schaltungshazards

Definition Schaltzungshazard Funktion hat bezüglich a, b keinen Hazard, Schaltnetz aber schon

Fragen

1. Wie kann das passieren?
2. Kann man das vermeiden?

Definition Schaltungshazard

Definition

Schaltnetz S für f hat einen statischen Schaltungshazard, wenn es $a, b \in \{0, 1\}^n$ gibt, so dass f bezüglich a, b keinen statischen Funktionshazard hat, aber beim Eingabewechsel von a nach b am Ausgang von S nicht notwendig stabil $f(a)$ anliegt.

Definition

Schaltnetz S für f hat einen dynamischen Schaltungshazard, wenn es $a, b \in \{0, 1\}^n$ gibt, so dass f bezüglich a, b keinen dynamischen Funktionshazard hat, aber beim Eingabewechsel von a nach b am Ausgang von S mehr als ein Funktionswertwechsel auftreten kann.

Beachte: Auftreten nicht garantiert!

Beispiel statischer Schaltungshazard

$$f(x_1, x_2, x_3) = x_1\, x_2 \vee \overline{x_2}\, x_3$$

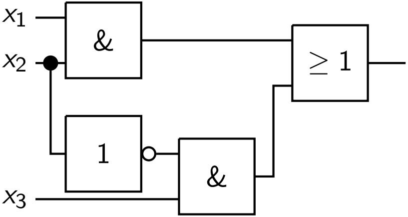
x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

$a = 111,\ f(a) = 1$

$b = 101,\ f(b) = 1$

klar kein Funktionshazard

Gar kein c zwischen a und b!



Vermeidung von Schaltungshazards

Wir konzentrieren uns auf

- statische Schaltungshazards,
- Schaltnetze, die direkt Polynome realisieren.

Sind in solchen Schaltnetzen Schaltungshazards ganz vermeidbar?

gute Nachricht ja

schlechte Nachricht nicht kostenlos

Wie vermeidet man statische Schaltungshazards?

Satz

Sei S ein Schaltnetz, das direkt ein Polynom p einer booleschen Funktion f realisiert. S hat genau dann keinen statischen Schaltungshazard, wenn S für jeden Primimplikanten von f ein Und-Gatter enthält.

Beweis durch Widerspruch:

statischer Schaltungshazard $\Leftrightarrow$ ein Primimplikant fehlt

Beweistrategie

1. Beweise: statischer Schaltungshazard $\Rightarrow$ ein Primimplikant fehlt
2. Beweise: Primimplikant fehlt $\Rightarrow$ statischer Schaltzungshazard

Hazard $\Rightarrow$ Primimplikant fehlt

Voraussetzungen

- Es gibt statischen Hazard bezüglich a und b.
- Es gibt keinen Funktionshazard bezüglich a und b.
- a und b unterscheiden sich in genau einem Bit

1\. Fall $f(a) = f(b) = 0$

klar alle Und-Gatter permanent 0

also Wert am Ausgang permanent 0

also kein Hazard $\surd$

2\. Fall $f(a) = f(b) = 1$

Sei m_a Minterm zu a, m_b Minterm zu b

Beobachtung $\exists$Monom m, Variable x: $\{m_a, m_b\} = \{m\,x, m\,\overline{x}\}$

klar m ist Implikant von f

also S enthält Und-Gatter für Verkürzung m' von m

Beobachtung Dieses Und-Gatter ist permant 1 (da unabh. von x).

also Wert am Ausgang permanent 1 also kein Hazard $\surd$

Primimplikant fehlt $\Rightarrow$ Hazard

Voraussetzungen

- S realisiert direkt Polynom p für f
- $\exists$Primimplikant m von f: kein Und-Gatter für m in S

Definiere Eingaben a und b.

- $x_i \in m \rightsquigarrow a_i = b_i = 1$
- $\overline{x_i} \in m \rightsquigarrow a_i = b_i = 0$
- sonst $\rightsquigarrow a_i = 0,\ b_i = 1$

Beobachtung $m(a) = m(b) = 1$

also $f(a) = f(b) = 1$

Beobachtung kein Hazard $\Leftrightarrow$ ein Und-Gatter permanent 1

klar So ein Und-Gatter realisiert Verkürzung von m.

also So ein Und-Gatter gibt es in S nicht.

also Hazard $\square$

Zusammenfassung: Hazards

Fazit Hazards

- Hazards sind ärgerlich.
  Wann liegt der richtige Funktionswert vor?
- Funktionshazards sind so nicht vermeidbar.
- Statische Schaltungshazards können mit zusätzlichem Aufwand vermieden werden.
- Eine grundsätzliche Lösung ist wünschenswert.

mögliche grundsätzliche Lösung: Taktung der Schaltung $\Rightarrow$ später