Entwurfsmethodik
Institut für Informatik

Modellierung stetiger Degradationseffekte in analogen Schaltungen

Felix Salfelder   25. September 2012

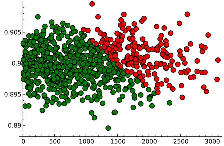
GOETHE UNIVERSITÄT
FRANKFURT AM MAIN

Schalterung und Robustheit

- Kontext, Ziele
- Stetige Degradation vs. unstetige
- Regelbare Zerfallsprozesse
- Ein Genetischer Algorithmus
- Anwendung auf Transistorschaltungen
- Hierarchische Modellgenerierung

Zusammenhang mit Robustheit, Ziele

- Intuition: Robustheit hängt mit Verhaltensparameteränderung zusammen.
- Verhalten über Zeit interessant, in Abhängigkeit von Störungen oder Umgebungsbedingungen oder (stressbedingter) *Alterung*.
- Funktioniert eine Schaltung nach einer *Mission* noch wie spezifiziert?



- Ziel: Alterungseffekte auf Systemebene
  Modellierung von (stetigen) stressabhängigen Parametern
- Und: Fitten des Modells an Mess- oder Simulationsdaten

Alterungseffekte, Überblick

- unstetige, TDDB, EM
- *stetige*, HCI, NBTI, PBTI

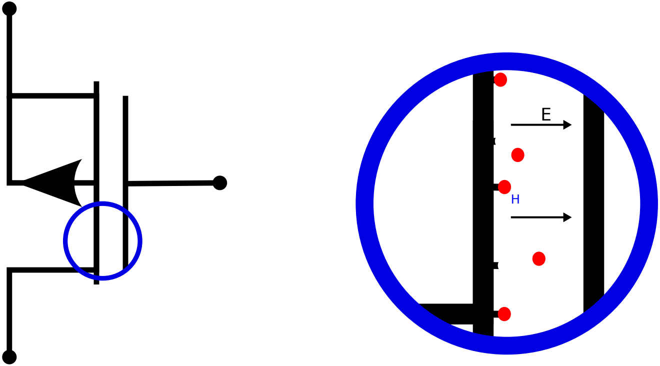
Unterscheidung wichtig, da Modellierungskonzepte tendentiell unterschiedlich.

- unstetig: Statistik (etwa: Black's Law)
- stetig: Physik (etwa: reaction-diffusion)

Heute: Hybridmodell für stetige Effekte

- analog
- ausheilungsfähig
- schnell
- datenbasiert

BTI physikalisch



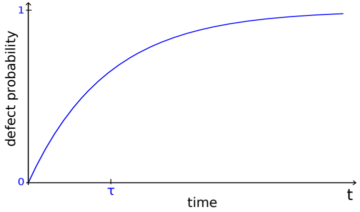
- E-Feld verantwortlich für Schwellspannungsverschiebung
- Effekt unstetig: H wird frei, V_{th} springt (ggf. rückwärts)
- Physikalisches (reaction-diffusion) Modell stetig (aber zu rechenintensiv).
- Idee: Modelliere den Beitrag zu ΔV_{th} eines jeden Hs.
- Statistik der Hs stetig.

Zerfallsprozesse

- Stochastischer Prozess ist zeitabhängige Zufallsvariable X
- Hier: $X_t : 0, 1 \to \mathbb{R}$ modelliert den Zustand eines Hs zur Zeit t.
- Erwartungsswert $\mathrm{E}(X_t)$ stetig in t.
- Beispiel: $X_0 = 0$.
  $P(X_{t+\delta t} = 1) = P(X_t = 1) + P(X_t = 0) \cdot \delta t / \tau$
- Exponentieller Prozess: *Konstante* Zerfallsdichte. Anschauung:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{E}(X) \sim E_0 - \mathrm{E}(X)$$

- Lösung mit Zeitkonstante τ: $\mathrm{E}(X_t) = E_0 \cdot (1 - \exp(t/\tau))$

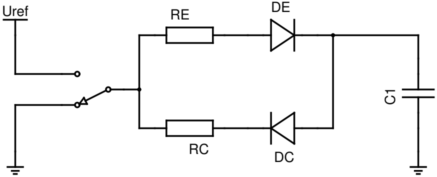


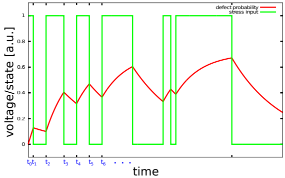
Digitales BTI Modell

- Ausheilung einfach im digitalen Fall.
  $P(X_{t+\delta t} = 1) =$

$$\begin{cases} P(X_t = 1) + P(X_t = 0) \cdot \delta t / \tau_\mathrm{e}, & \text{falls an} \\ P(X_t = 1) \cdot (1 - \delta t / \tau_\mathrm{c}), & \text{falls aus} \end{cases}$$

- BTI digitales *H* Modell (Schlünder & alii)





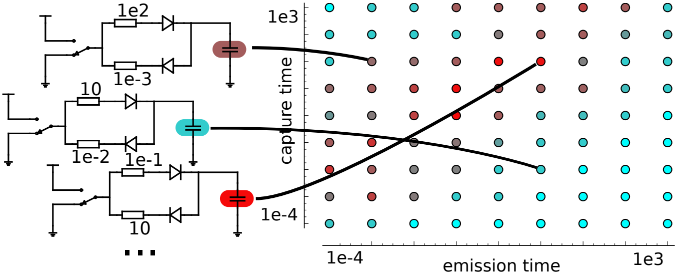
- Korrespondenz zur Anschauung:
  Kondensatorspannung $\Leftrightarrow$ Erwartungswert von X_t

Digitalmodell explizit

- Alterungsparameter ist gewichtete Summe von Erwartungswerten

$$\Delta P = \Delta V_{\mathrm{th}} = \sum w_i \cdot E(X_i)$$

- Hier: zwei *konstante* Zeitkonstanten pro *Zelle*

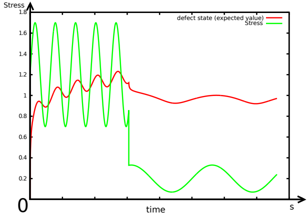


Analoges Modell

- Fallunterscheidung nach Stresspegel s funktioniert so nicht.
- Idee: beide Prozesse sind gleichzeitig aktiv, mit *variabler* Zerfallsdichte/Zeitkonstante

$$\begin{aligned} P(X_{t+\delta t} = 1) = & P(X_t = 0) \cdot \delta t / \tau_{\mathrm{e}}(s) \\ & + P(X_t = 1) \cdot (1 - \delta t / \tau_{\mathrm{c}}(s)) \end{aligned}$$

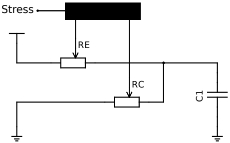
- beliebiger Stresspegel zulässig, Erwartungswert wird um Gleichgewicht zentriert.



- abwärtskompatibel zu BTI-digital Modell (EdaWorkshop 11)
- somit prinzipiell sinnvoll

Analogmodell explizit

- Stress steuert Zeitkonstanten, Abhängig von der Wahl der Prozesse

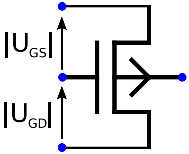


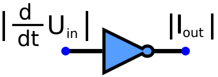
- Viele Zellen liefern Erwartungswerte ($\hat{=}$ Kondensatorspannugen)
- Alterungsparameter wieder gewichtete Summe dieser (zentrierten) Erwartungswerte

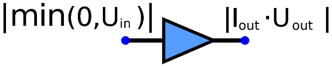
$$\Delta P = \sum w_i \cdot \Delta \mathrm{E}(X_i)$$

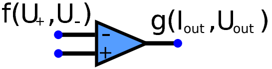
- Noch benötigt: Stresspegelmodell und Auswahl der Zellen

Stresspegelmodell

- (zeitvariable) Größe, mit der Alterung einhergeht
- von *aussen* sichtbar (ohne Netzliste/physikalisches Modell)
  - BTI 

  - HCI (in Inverter) 

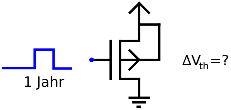
  - Endstufe, zum Beispiel 

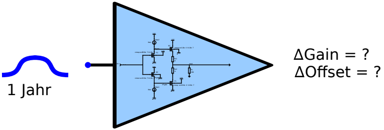
  - Differenzverstaerker 

- Einfach für kleine Schaltungen, sonst schwieriger.
- Noch nicht automatisiert...

Parametersuche für Alterungsmodell

- bekannt: Stressignale $s_1, \ldots, s_n$, Parameter $p_1, \ldots, p_m$.
- Zum Beispiel: Messung oder Simulation





- gesucht: Modell, das die p_i aus den s_i reproduziert. Parametrisiert durch
  - Menge von Zellen
  - ... für jedes der Signale s_i
  - τ_e und τ_c für jede Zelle
  - Gewichtung der Zellen.
- Dann: Parameterverschiebung kann eine Ebene höher simuliert werden

Genetischer Algorithmus

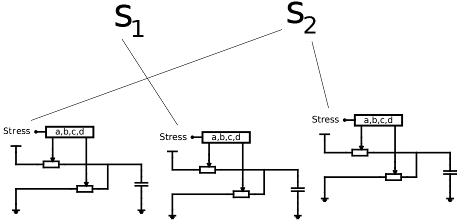
- Genom legt Suchraum fest. Punkt im Suchraum: *Individuum*
- Kreuzung, aus zwei Individuen mach zwei Nachkommen
  hier: durchmischen der Zellen
- Mutation, zufällige Reparametrisierung
- Fitness Funktion, bewertet Individuum.
- randomisierte Suche nach Fit.

Genetischer Algorithmus überwindet numerische Schwierigkeiten.

- Geschlossene Lösung für Erwartungswerte nicht immer vorhanden
- und wenn: Ableitungen füllen mehrere Bildschirmseiten
- ... sind numerisch instabil.

Genetischer Algorithmus – konkret

- Genom: feste Anzahl von Zellen, etwa zehn.
- Parametrisierung der Zeitkonstantenabhängigkeit von Stresspegel s
  Alterung/Verschlechterung: $\tau_e = \exp(a \cdot s + b)$
  Ausheilung/Verbesserung: $\tau_c = \exp(c \cdot s + d)$
- Wichtig: $a \geq 0 \geq c$. Damit mehr Stress zu mehr Schaden führt!
- Jeder Zelle wird ein Stresssignal zugeordnet
- Somit: jede Zelle hat fünf Parameter, zusammen 40 Zahlen und 10 Zuordnungen.

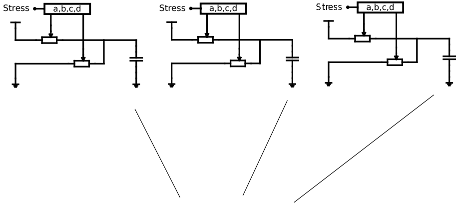


- (Die Gewichte w_i ergeben sich aus den Messdaten)

Fitness

Problem: Wie gut approximiert ein Individuum die Messwerte?

- Für alle Paare s_i, X_j auswerten: Große Matrix E.
  (Falls s_i rechteckig: Prozess besitzt geschlossene Lösung!)
- Optimale Gewichte $g = g_0, ..., g_{10}$ berechnen: Quadratische Optimierung:



$$\sum (m_i - (E \cdot g)_i)^2 \text{ minimal und } g_i \geq 0$$

- Abstand Modell zu Messdaten ist genau die Zielfunktion dieser Optimierung.

Simulation

- Integration von BTI in BSIM3 Modell als Zerfallprozessmodell

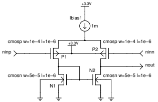
```
.model bti_ptm bti_sum (
  parms = (
  (1.2893, 0.0, 13.433, 43.57692, 1.4509),
  ...
  (25.554, -46.32138, 16.8444, 11.97632, 23.2963)))
.model mypmos pmos ( level=8 ...  bti=bti_sum )
```

- Steuerung einer Zwei-Zeiten-Simulation

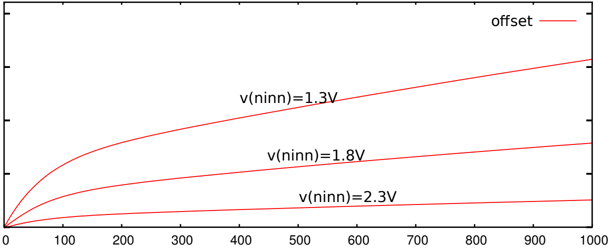
```
.include netlist.ckt
.print tran v(out)
.tran 0 1m 1u > fresh
.tw 1m 1u 100 tran > aging
.tran 0 1m 1u > aged
.tw 100 pd > powerdown
.tran 0 1m 1u > annealed
```

Simulation Beispiel 1

- Komparator, asymmetrische Belastung. `v(ninp)` Sinusförmig.

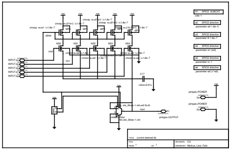


- Beobachtung: Offsetverschiebung hängt von Betriebsmodus ab

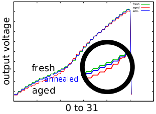


Simulation Beispiel 2

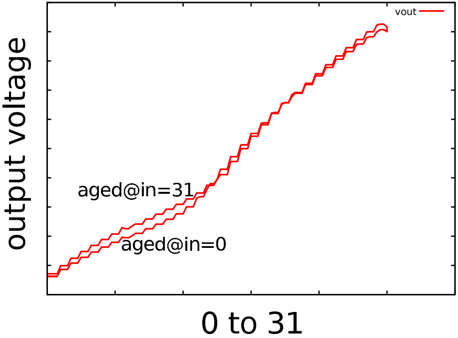
- Stromgesteuerter Wandler



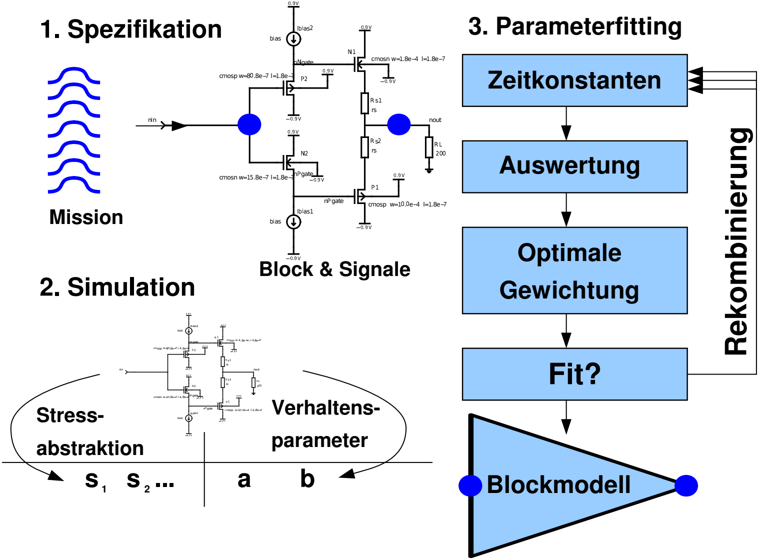
- Ausheilung



- Betriebsmodusabhängigkeit



Hierarchischer Flow



Blockmodell mit Alterungsparameter

```
module amp (out, gnd, dd, in);
...
  electrical out, gnd, dd, in, s0, s1;
  degradational p1, p2, ... p10;
  // submodules
  decay3 #(-33.03, 2.048, 11.518,-21.489) d0 (s0, 0, p1):
  decay3 #(-7.672, 8.063, 40.824, 23.874) d1 (s1, 0, p2);
  ..
  decay3 #(-19.97, 20.25, 28.564,-10.145) d10(s1, 0, p10);
  analog begin
    real offset=231.4*State(p1) + 20123.5*State(p2) + ... ;
    I(out,gnd) <+ Verhaltensmodell( V(in), offset, ... );
    // Stress evaluation
    Stress(stress0) <+ Funktion( V(in), I(out), ...);
    ...
  end
endmodule
```

Letzte Folie

Vielen Dank!