

Aktuelle Entwicklungen bei Place and Route Algorithmen

Stephan Larws

Dresden, 24. Juni 2009

Gliederung

- 1 Place & Route
- 2 Min-Cut Platzierung
- 3 Kräfteplatzierung
- 4 Simulated Annealing (SA)
- 5 Genetische Algorithmen (GA)
- 6 Kombination von GA und SA

1 Place & Route

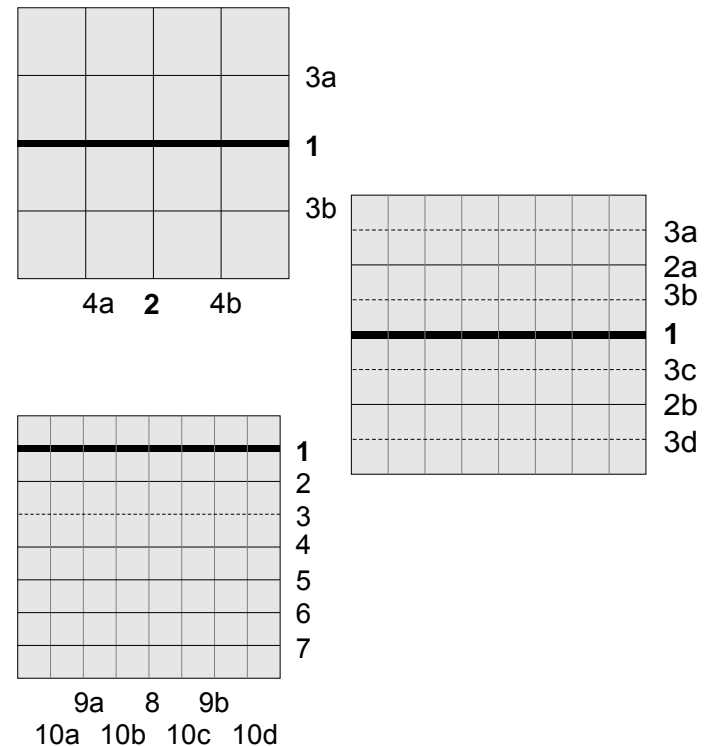
- Ein Schritt in der Schaltungsentwicklung, Teil der Layoutsynthese.
- Hierbei erfolgt die Platzierung und Verdrahtung der Bauelemente.
- Bei FPGAs aufgrund der Komplexität fast ausschließlich automatisiert.
- Ziel ist es meist die optimale Platzierung bei möglichst geringer Verdrahtung zu erreichen.
- Andere Möglichkeiten der Optimierung (Laufzeiten, Fläche, Verdrahtungslänge, maximale Schnittzahl, ...).

2 Min-Cut Platzierung

- Optimierungskriterium ist hierbei die Verbindungsschnitte zwischen den einzelnen Zellen zu minimieren und somit die Gesamtverbindungslänge zu reduzieren.
- Erfolgt mit Hilfe des Kernighan-Lin Algorithmus (KL-Algorithmus) oder des Fiduccia-Mattheyses Algorithmus (FM-Algorithmus).
- Beide Algorithmen spalten einen Graph in zwei Teilgraphen mit möglichst geringen Schnittkosten.
- Nachteil ist, dass hierbei die Schnittkosten als einziges Kriterium nutzbar ist.

Min-Cut Platzierung (2)

- Verschieden Herangehensweisen für Schnittmuster:
 - Quadratur-Platzierung
 - Halbierungs-Platzierung
 - Reihen-/Halbierungs-Platzierung



3 Kräfteplatzierung

- Modellierung der Schaltung als ein mechanisches System mit Federn.
- Die Zellen entsprechen den Körpern und die Verbindungen dazwischen den Federn.
- Die Kraft verhält sich dabei proportional zu der Entfernung zwischen zwei Körpern.
- Gesucht sind die Positionen der Zellen mit dem geringsten Kräftegleichgewicht (ZFT-Position – Zero force target Position).

Kräfteplatzierung (2)

Skizzierung eines einfach Algorithmus zur Kräfteplatzierung

1. Erstellen einer zufälligen Anfangsplatzierung.
2. Auswahl einer Zelle, Berechnung der ZFT Position,
 - wenn ZFT frei, verschieben der Zelle.
 - wenn ZFT belegt, Wahl einer von vier Belegungsoptionen.
3. Weiter mit 2. bis Abbruchkriterium erreicht (bspw. Anzahl Iterationen,...).

Mögliche Belegungsoptionen, wenn ZFT belegt

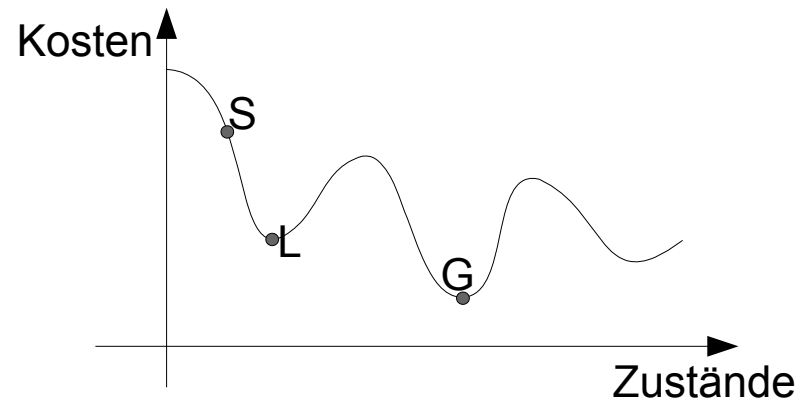
- Verschieben der Zelle möglichst nahe zur ZFT.
- Berechnen der Kostenveränderung bei Tausch beider Zellen. Bei geringeren Kosten, vertauschen beider Zellen.
- „chain move“: Verschieben der Zelle an berechnete Position, dortige Zelle wird an nächstliegende Position verschoben, ist diese belegt, wiederholtes Verschieben der dortigen Zelle.
- „ripple move“: Zelle wird auf ZFT verschoben, für die dortige Zelle neue ZFT berechnen. Solange durchführen bis alle Zellen platziert sind.

4 Simulated Annealing

- Basiert auf dem Abkühlungsprozess von geschmolzenem Metall.
- Ziel ist dabei die Erzeugung eines Metalls mit energieminimalen Atomgitters.
- Erfolgt durch schrittweise Senkung und nach jedem Schritt halten der Temperatur.
- Je höher die Temperatur, desto beweglicher sind die Metallatome → große Positionsveränderungen sind möglich.
- Bei sinkender Temperatur fallen die Bewegungsmöglichkeiten der Atome → Gitterstruktur entsteht.

Simulated Annealing (2)

- Auswahl eines zufälligen Startzustandes (S).
- Bei Erreichen eines lokalen Minimums (L) ist es meist möglich dieses wieder zu verlassen.
- Erfolgt durch Einführung eines Zufallswertes in Abhängigkeit von der Temperatur.



Simulated Annealing (3)

```
begin
  T = T0
  P = init_placement
  repeat
    repeat
      NewP = PERTURB(P)
      Δcost = COST(NewP) - COST(P)
      if(Δcost < 0) then
        P = NewP
      else
        r = RANDOM(0,1)
        if(r < e-Δcost/T) then
          P = NewP
    until(Abbruchkriterium, z.B. Gleichgewicht bei T)
    T = α * T      /* 0 < α < 1 */
  until(T < Tmin)
end
```

Quelle: [1]

- Komplexität im Wesentlichen nur von Vertauschungs- und Kostenfunktion abhängig.
- Abbruchkriterium der inneren Schleife hängt von der jeweiligen Implementierung ab.
- Weiterer Optimierungspunkt ist die Einstellung der Schrittgröße α .

5 Genetische Algorithmen (GA)

- Vermehrte Anwendung für Suchen in sehr großen Suchräumen.
- Begriffe im Zusammenhang mit GA:
 - **Chromosom**: Mögliche Lösung des Problems
 - **Fitness-Funktion**: Bestimmt die Güte einer Lösung
 - **Population**: Menge von möglichen Lösungen
 - **Generation**: Population, die durch Reproduktion entstanden ist
- Bestandteile eines einfachen GA:
 - Zufällig erstellte Anfangspopulation
 - Fitness-Funktion
 - 3 Operatoren: Selektion, Kreuzung und Mutation

Genetische Algorithmen (2)

- **Selektion:** Wählt Chromosomen für die Reproduktion aus. Je höher die Fitness des Chromosom, desto größer die Chance, dass es zur Reproduktion gewählt wird.
- **Kreuzung:** Wählt eine zufällig Stelle in zwei Chromosomen und kreuzt sie ab dieser.
Bsp.: zwei 8 Bit Chromosomen *00100011* und *11011110* werden ab der vierten Stelle gekreuzt. Es entstehen die beiden neuen Chromosomen *00101110* und *11010011*.
- **Mutation:** Verändert eine zufällige Stelle in einem Chromosom.

Genetische Algorithmen (3)

1. Beginne mit einer zufälligen Population von n Chromosomen.
2. Berechne die Fitness jedes Chromsomen.
3. Führe folgende Schritte solange durch, bis n Nachkommen erzeugt wurden.
 - a. Selektion zweier Elternchromosomen.
 - b. Kreuze mit Wahrscheinlichkeit p_c die beiden Elternchromosomen an einer zufälligen Stelle und erzeuge so zwei Kindchromosomen.
 - c. Mutiere mit Wahrscheinlichkeit p_m die beiden Kindchromosomen Falls n ungerade, verwerfen eines Kindchromosoms.
4. Ersetze die alte Population mit der neuen Generation.
5. Gehe zu Schritt 2.

6 Kombination von GA und SA

- Genetische Algorithmen konvergieren langsam.
- SA konvergiert schnell zum lokalen Optimum, kann dort aber auch hängen bleiben.
- Bei Kombination beider heben sich die Nachteile gegenseitig auf.
- Grober Umriss des vereinten Algorithmus:
 - Zu Beginn Anwendung des GA in der so genannten globalen Suchphase.
 - Nach einigen Generationen Wechsel zu SA, die lokale Suchphase.
 - SA übernimmt somit das „fine-tuning“.

Vielen Dank für die Aufmerksamkeit

Quellen

- [1] J. Lienig, Layoutsynthese elektronischer Schaltungen, Springer 2006
- [2] M. Mitchell, An Introduction to Genetic Algorithms, MIT Press 1996
- [3] V. Betz, J. Rose, Architecture and CAD for Deep-Submicron FPGAs, Kluwer Academic Publishers 2002
- [4] Yang Meng, A.E.A Almaini, Wang Pengjun, FPGA placement optimization by two-step unified genetic algorithm and simulated annealing algorithm, Journal of Electronics (China) 07/2006