

COMPONENT INTERCONNECTION INFERENCE ALGORITHM

Nela Olšarová

Master Degree Programme (2), FIT BUT

E-mail: xolsar00@stud.fit.vutbr.cz

Supervised by: Zbyněk Křivka

E-mail: krivka@fit.vutbr.cz

Abstract: This paper deals with the design of hardware component interconnection inference algorithm. The algorithm is designed to be used in the FPGA schema editor to support user by finding optimal interconnection of two given components. The clue information about components (interface) are stored in the knowledge base. The inference of the best possible interconnection is related to the bipartite matching and assignment problem. After the discussion of this relation, we present experimental results obtained by a testing of the algorithm with several small FPGA designs.

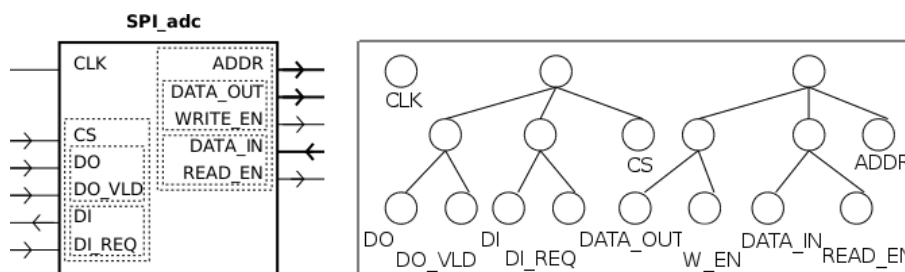
Keywords: FPGA, assignment problem, bipartite matching, inference, component, interface, port

1 ÚVOD

Zde popsáný algoritmus pro inferenci propojení hardwarových komponent je určen pro použití ve vznikajícím editoru návrhů pro FPGA. Má usnadnit uživateli editoru vytváření schémat zapojení. Uživatel vybere dvě komponenty k propojení, algoritmus poté navrhne několik nejlepších variant a uživatel z nich na konec vybere tu nejhodnější, kterou může ještě ručně upravit. V dalším textu bude detailněji popsán tento algoritmus a dílčí problémy, které je potřeba při jeho návrhu řešit. V závěru bude zhodnocena úspěšnost aktuální implementace při testování na sadě malých návrhů.

2 NÁVRH INFERENČNÍHO ALGORITMU

Vstupem pro algoritmus jsou dvě uživatelem vybrané komponenty. *Rozhraní* každé komponenty je popsáno v existující bázi znalostí a je tvořeno hierarchickou strukturou *portů* a *pinů* (listový port šířky jedna), jak naznačuje obrázek 1. Každý port komponenty má dále vlastnosti, které určují jeho šířku a kompatibilitu s jiným portem, jako směr (např. nelze propojit dva výstupní porty) nebo typ (datový, adresový, řídicí). Každá vlastnost je zároveň odvozena podle listových portů i pro obsahující nadporty. Porty mohou mít také již částečně obsazenou šířku, pokud se komponenta účastní ve vytvářeném schématu také v propojení s jinými komponentami.



Obrázek 1: Hierarchická struktura portů komponenty SPI_adc (je vynechán kořenový uzel zastupující obsahující komponentu).

Výstupem algoritmu je několik nejlepších propojení (množin párů listových portů) vstupních komponent seřazených sestupně podle ohodnocení. Mimo to zároveň dojde k určení potřebných výsledných šířek pro konkrétní odvozené propojení z dosud nerozhodnutých generických parametrů pro šířky portů.

2.1 SOUČÁSTI INFERENČNÍHO ALGORITMU

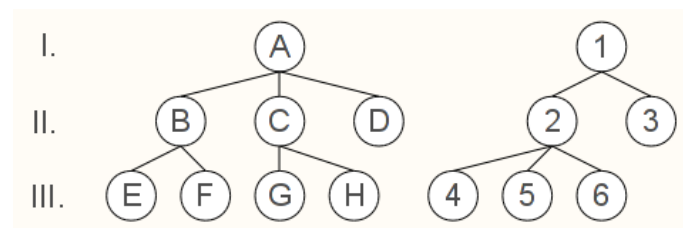
Inferenční algoritmus je složen z řešení několika dílčích problémů. Nejvýznamnějším je *hledání nejlepšího párování portů*, kdy hledáme nejlepší párování mezi dvěma disjunktními množinami prvků v závislosti na jednotlivých ohodnoceních vytvářených párů. Jedná se o teoreticky časově nejnáročnější část algoritmu. Tento kombinatorický problém bude detailněji rozebrán v části 2.2.

Elementární operací při párování je *test kompatibility* dvojice portů podle jejich vlastností, tedy zda je možno tyto dva porty propojit. *Ohodnocovací funkce* ohodnotí propojení dvou kompatibilních portů, je využívána při hledání nejlepšího párování portů. *Výpočet šířek propojení* je aplikován v závěru algoritmu pro zjištění využitých kapacit šířek komponent a určení nerozhodnutých generických parametrů pro šířku portů. Celý algoritmus také bere v úvahu dosavadní obsazení portů již provedeným propojením.

2.2 PROPOJOVÁNÍ PORTŮ KOMPONENT Z POHLEDU BIPARTITNÍHO PÁROVÁNÍ A PŘÍRAZOVACÍHO PROBLÉMU

S řešením podproblému nalezení nejlepšího párování portů na základě ohodnocovací funkce úzce souvisí problematika bipartitního párování (*Bipartite Matching, BM*) [1] a přiřazovacího problému (*Assignment Problem, AP*) [2], který je jeho specializací. Algoritmy, které párování řeší, patří do rodiny síťových algoritmů (*Network Algorithms*). Algoritmy pro řešení základních podob přiřazování a párování leží ve třídě časové složitosti P [2].

Zkoumání výše uvedených problémů a jejich modifikací, které již zpravidla leží ve třídě časové složitosti NP [2], jako kvadratický (*Quadratic AP*) nebo zobecněný přiřazovací problém (*Generalized AP, GAP*), vedlo k vytyčení možných způsobů, jak efektivně řešit hledání nejlepšího propojení portů dvou komponent. Uvažujme například stromové struktury portů vstupních komponent, které jsou znázorněny na obrázku 2.



Obrázek 2: Příklad struktury dvou vstupních komponent.

Při aktuálně implementovaném přístupu *shora dolů* (tedy započítáním párování od kořenů stromových struktur) nejprve vytvoříme jednoznačný pár A-1 a poté hledáme rekurzivně páry mezi přímými potomky. Pro pár B-2 vzniklý při dalším sestupu by to znamenalo prohledávání mezi množinami {E, F} a {4, 5, 6}. Na základě sumy ohodnocení nejlepší kombinace listových párů ohodnotíme i pár přímých předků, a tak jsme na každé úrovni schopni řešit tzv. maximální váhované bipartitní párování (*Maximum Weighted BM*) [1], které je řešitelné v polynomiálním čase.

Další sledovaný přístup byl *zdola nahoru*. Zde vstupuje do ohodnocovací funkce závislost ohodnocení

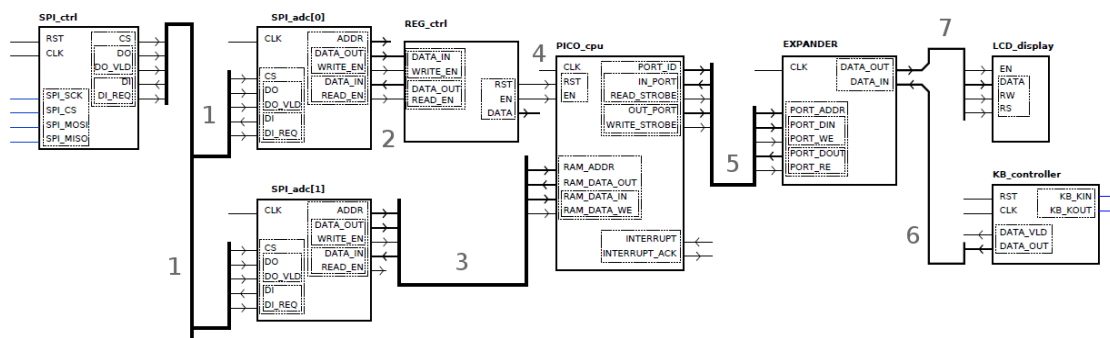
jednoho páru na jiných zároveň vytvořených párech, což činí z příbuzného přiřazovacího problému kvadratický, který není řešitelný v polynomiálním čase [2], a tak by byla potřeba zavést některou heuristiku nebo jiné metody umělé inteligence uváděné v [2].

Algoritmus je také omezen nemožností vyhledat propojení jednoho portu jedné komponenty na více portů druhé komponenty zároveň. Toto by vedlo na GAP, který je opět ve třídě NP [2].

3 IMPLEMENTACE A TESTOVÁNÍ ALGORITMU

Editor návrhů pro FPGA je vytvářen jako plugin pro vývojové prostředí založené na prostředí Eclipse, implementačním jazykem projektu a také obsaženého inferenčního algoritmu je jazyk Java.

Algoritmus byl testován na šesti aplikacích z výuky, při jejichž propojování nalezne hledané propojení pro 90 % obsažených dvojic komponent. Neúspěch algoritmu v 10 % případech je dán výše zmíněným GAP omezením a nepřesností v situaci, kdy hledaným propojením je pouze podmnožina nejlépe ohodnoceného propojení. Ilustrujme úspěšnost algoritmu na aplikaci na obr. 3.



Obrázek 3: Picoblaze programovatelný z SPI s LCD a klávesnicí.

Algoritmus v průběhu testování našel úspěšně všechna znázorněná propojení, kromě č. 7, které vyžaduje nalezení propojení jednoho portu na více portů druhé komponenty zároveň (nalezl pouze částečné řešení). Tento a příbuzné případy bude potřeba explicitně zapsat do báze znalostí a následně toto propojení místo odvození obecným inferenčním algoritmem rovnou nabídnout uživateli.

4 ZÁVĚR

Algoritmus pro inferenci propojení komponent byl realizován přístupem shora dolů a ve své stávající podobě úspěšně nalézá všechna očekávaná propojení. V plánu je vylepšení časové složitosti, kdy bude již využit polynomiální algoritmus pro řešení maximálního váhovaného bipartitního párování.

5 PODĚKOVÁNÍ

Tato práce byla podpořena projektem MŠMT 2C06008 „Virtuální laboratoř aplikace mikroprocesorové techniky“ a výzkumným záměrem MSM 21630528.

REFERENCE

- [1] Mehlhorn K., Naher St.: The LEDA Platform of Combinatorial and Geometric Computing, Cambridge University Press, 1999
- [2] Pentico David W: Assignment problems: A golden anniversary survey, Graduate School of Business, Duquesne University, Pittsburgh, PA 15282-0180, USA, 2005