

Implementace genetického algoritmu do obvodu FPGA

Petr Burian

Katedra aplikované elektroniky a telekomunikací, Západočeská univerzita v Plzni

Implementation of Genetic Algorithm by FPGA Circuit

Abstract

This paper deals with the implementation of a standard genetic algorithm by an FPGA circuit. It examines the various features of this algorithm. The main goal of this work is building of an evolvable combinational circuit. Demands imposed on an FPGA circuit are researched as well.

Keywords

Genetic algorithm, evolvable hardware, SGA.

Úvod

Skupina evolučních algoritmů, kam patří také genetický algoritmus, tvoří nástroje, jimiž lze velmi efektivně řešit širokou skupinu optimalizačních problémů a úloh. Jak název napovídá, jedná se o skupinu algoritmů, které jsou založeny na evoluci. Tedy na principu postupného vývoje, který více či méně přebírá mechanismy vývoje druhů v přírodě. Pokud se tyto optimalizační metody spojí například s rekonfigurovatelným číslicovým či analogovým obvodem, vznikne obvod, který je v jistých mezích schopen samostatného vývoje. Tato problematika, která se v posledních době stává velmi aktuální, se nazývá evoluční hardware.

Standardní genetický algoritmus

Jedná se o základního zástupce evolučních výpočetních technik. SGA probíhá v několika krocích. Jako první se inicializuje počáteční populace jedinců. Jedinec je v SGA typicky reprezentován chromozomem ve tvaru binárního řetězce o konstantní délce. Každý jedinec zároveň představuje právě jedno řešení optimalizační úlohy. Skupina jedinců pak tvoří populaci. Nejjednodušší inicializace se provádí tak, že binární řetězce jedinců se vyplní náhodnými hodnotami. Zásadní částí algoritmu jsou pak tzv. rekombinační operátory, jsou to operátory, které se aplikují na jedince či na jejich skupiny a tak pozměňují jejich funkci při realizaci úlohy. Tyto operátory existují obvykle dva – křížení a mutace.

Křížení může probíhat různými způsoby. Záleží na řešeném problému. Pro SGA je ale nejtypičtější tzv. jednobodové křížení. Pravděpodobnost, se kterou se dostane jedinec do procesu křížení je dána tzv. pravděpodobností křížení (P_k), která bývá nejčastěji kolem 0,8. Další rekombinační operátor, kterým je mutace, způsobuje menší zásahy do genetické informace jedince. Je vyjádřen též pravděpodobností – pravděpodobností mutace ($P_m =$ cca 0,15). Často se realizuje tak, že nad každým bitem jedince se provede náhodný pokus a podle toho se pak daný bit ponechá buď v přímém, nebo v negovaném tvaru. Existují i varianty kdy se náhodně znekuje v jedinci určitý počet bitů.

Po změně jedinců pomocí rekombinačních operátorů získáváme kromě jedinců předchozí generace ještě jejich potomky a mutanty. Aby bylo možné určit, kdo přejde (proces selekce) do další generace, je nutné jedince ohodnotit podle kvality řešení, které přinášejí. Cílem je tzv. hodnota fitness pro každého jedince. Pro výběr jedinců do další generace existují dva nejčastější postupy, tzv. ruleta a turnaj. Turnaj ani ruleta nám ale nedává

jistotu, že nejlepší jedinec přežije (zjevná analogie z přírody), proto se často tyto selekční mechanismy doplňují tzv. elitismem. Tzn. prvkem, který vždy zajistí přechod nejlepšího jedince do další generace. Po selekci jedinců do nové generace se celý proces může znovu opakovat a to až do chvíle, kdy bude splněna buď podmínka na dostatečnou kvalitu řešení, nebo až do vyčerpání určitého počtu evolučních kol [1].

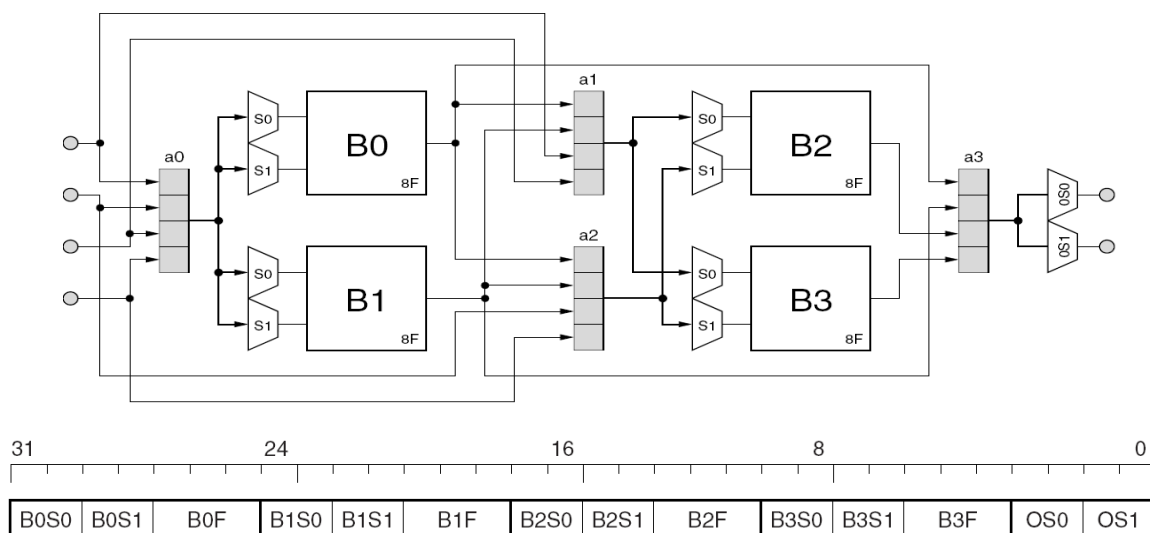
Hardwarová realizace v obvodu FPGA

Realizace SGA má ověřit, zda lze algoritmus v tomto obvodu reálně implementovat a zda je možné řešit evoluci v rozumných časových mezích. Implementaci do FPGA nahrává také fakt, že je možné část obvodu využít jako rekonfigurovatelný obvod a tak dospět k evolučnímu obvodu. Obvod pro rekonfiguraci, tedy prostředí, na kterém bude evoluce probíhat, byl převzat z [2]. Jedná se o jednoduchý obvod, který může být základním stavebním prvkem pro složitější obvodové struktury. Obvod se skládá ze čtyř buněk, které jsou schopné realizovat logické funkce NOR, $x \text{ AND } \text{not } y$, $\text{not } x \text{ AND } y$, AND, OR, $\text{not } x \text{ OR } y$, $x \text{ OR } \text{not } y$, NAND. Tyto buňky doplňují konfigurovatelné multiplexery, které řídí tok signálů mezi buňkami. Celkově je schopen obvod realizovat kombinační obvod se 4 vstupy a 2 výstupy. Konfigurace obvodu je v podobně 32-bitového řetězce. Ovšem počet realizovatelných obvodů není 2^{32} , jak by se dalo domnívat, ale jen 178764, a to proto, že některé konfigurace vedou ke kombinačním obvodům se stejnou pravdivostní tabulkou [2].

K realizaci byl zvolen obvod Altera Cyclone II. Cílem testovací aplikace bude získat evoluci kombinační obvod jen na základě zadané pravdivostní tabulky. Algoritmu tedy poskytneme jen očekávanou odezvu obvodu, a jeho vhodnou konfiguraci zajistí SGA. Pro tento test byla zvolena populace o velikosti 64 jedinců (každý jedinec 32bitů). Pro uchovávání jedinců, stejně tak jako jejich ohodnocení je využívána vložená (embedded) paměť v obvodu FPGA. Celý obvod realizující SGA je koncipován jako periférie pro soft. procesor NIOS II přes sběrnici Avalon. Toto řešení zajistí dobrou a snadnou kontrolu nad algoritmem a případně umožní i dynamickou změnu parametrů evoluce. Procesor evoluční kroky pouze spouští a vyhodnocuje, nepodílí se tedy na výraznějším zpomalování algoritmu, ten probíhá ve vytvořené periférii a tedy „na pozadí“ výpočetního času procesoru. Struktura periférie je navržena tak, že každá jednotlivá fáze algoritmu je realizována jedním automatem. Ty jsou postupně spouštěny pomocí „control registru“ (ovládán procesorem NIOS II), tento registr může být i čten, takže je možné získat dobrý přehled o tom, ve které fázi optimalizace se algoritmus nachází a především jak je která fáze časově náročná. Kromě těchto stavů může procesor přes sběrnici načíst i aktuálního leadera (nejlepší jedinec) a jeho ohodnocení, což je zásadní především pro ukončovací podmínku algoritmu.

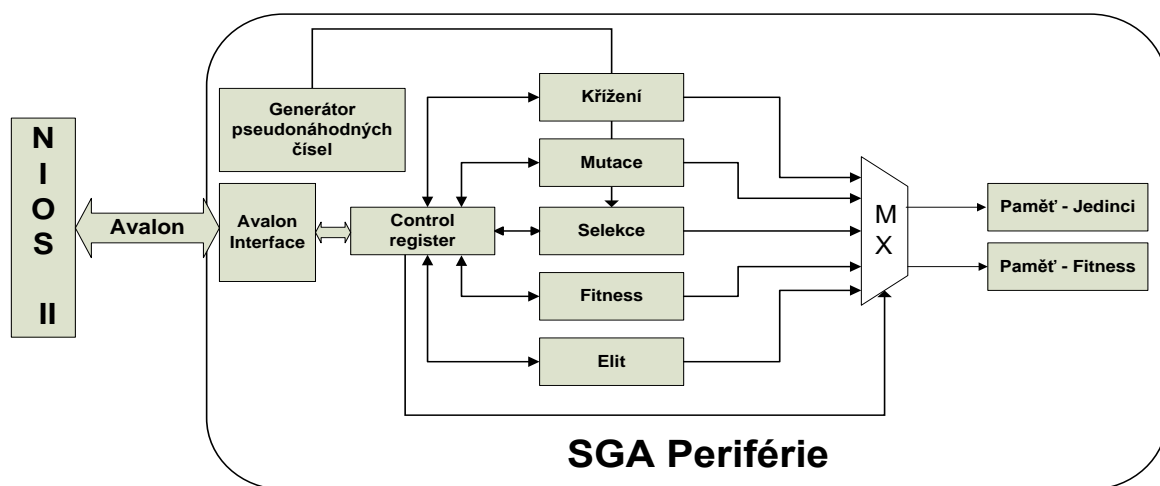
Pokud se jedná o modifikaci algoritmu, byla zvolena nejobvyklejší varianta SGA. Křížení je realizované jako jednobodové, což pro realizaci v FPGA nepředstavuje žádný problém. Složitější situace je u mutace. Pokud se provádí náhodný pokus u každého bitu, vyžaduje to tedy 32 náhodných pokusů nad jedním jedincem, což ale v případě sériového zpracování může vést na výrazný nárůst výpočetního času (minimálně 32×64 taktů), v případě paralelního na nárůst použité programovatelné logiky. Řešením může být například mutace jen jednoho, náhodně vybraného bitu u každého jedince. Tento přístup může skýtat výhody i nevýhody co se týče vlastní evoluce, náhodný test u každého bitu může přinést rychleji požadovaný výsledek, na druhou stranu, mutace jen jednoho bitu dokáže jedince měnit „jemněji“. Bylo rozhodnuto implementovat obě metody, a to jednoduše - sériově, a pak porovnat jednotlivé výsledky. Ohodnocení jedinců, které se musí po rekombinačních operacích provádět, je v tomto případě nasnadě. Jedinec je použit jako konfigurační řetězec pro rekonfigurovatelný obvod, následně je tento obvod otestován na všech 16 stavů (4-bitový vstup) a jeho výstupy porovnány s požadovaným ideálním výstupem (16x2 bity).

Fitness funkce může tedy maximálně nabývat hodnoty 32, což značí naprostou shodu s požadovanou výstupní odezvou. Pro selekci nové generace byl zvolen výběr turnajem. A to především pro jeho snadnou realizaci logikou. Jedná se o turnaj dvou náhodně vybraných jedinců, který se opakuje, dokud není nová generace naplněna. Turnaj vybírá z těchto jedinců: jedinci staré generace, potomci a mutanti. Turnaj tedy tvoří novou generaci (64 jedinců) z 192 (3x64) možných jedinců. Je doplněn elitismem, tzn. přežitím nejlepšího jedince, který se během evolučního kola objeví. Vzhledem k tomu, že SGA je založen na náhodných pokusech, je nutné získat generátor náhodných čísel. Ten je v FPGA vytvořen pomocí zpětnovazebního lineárního posuvného registru (LFSR).



Obr.č. 1: Rekonfigurovatelný obvod a jeho konfigurační řetězec

Zdroj: [2]



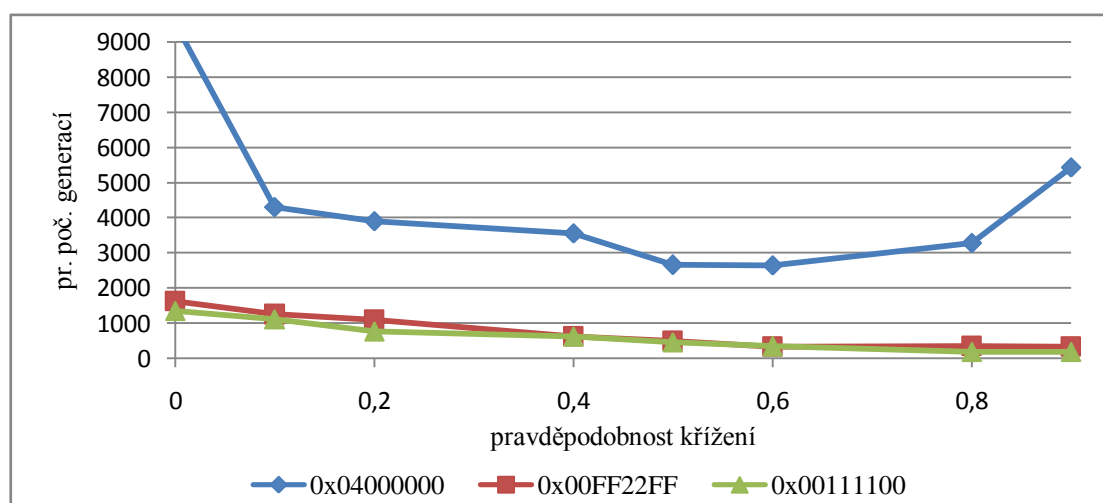
Obr.č. 2: Blokové schéma SGA periférie

Výsledky

Pro ověření činnosti evolučního obvodu bylo zvoleno několik náhodně vybraných pravdivostních tabulek. Evoluce byla spouštěna s různými hodnotami P_k a P_m , aby se ukázala závislost rychlosti evoluce na těchto parametrech. Měření probíhalo na obou variantách řešení mutace. Protože výsledky evoluce jsou vždy závislé na náhodě, byla každá evoluce spuštěná 200x, aby bylo možné získat relevantní statistické údaje. Pravdivostní tabulka ke které má algoritmus dospět se zadává v podobě 32-bitového čísla (prvních 16 bitů značí výstupní funkci prvního výstupu, dalších 16 bitů druhého výstupu).

vzorek	P_k	P_m	P_k (1-bit)	prům. poč. generací	prům. čas evoluce [ms]
0x04000000	0,6	0,15	-	2652,03	164,48
0x04000000	-	-	0,2	2388,18	82,18
0x00111100	0,5	0,05	-	146,34	8,58
0x00111100	-	-	0,4	169,74	6,17
0x00FF22FF	0,6	0,15	-	340,87	20,55
0x00FF22FF	-	-	0,3	334,74	11,59
0x5050F000	0,9	0,15	-	17,58	1,23
0x5050F000	-	-	0,3	18,40	0,69

Tab.č.1: Optimální P_m a P_k – jednobitová i vícebitová mutace



Graf č. 1: Závislost počtu generací na P_k při $P_m=0,15$

Graf č.1 ukazuje průměrný počet generací při proměnné P_k a při konstantní pravděpodobnosti mutace. Je zřejmé, že pokud se evoluce musí spolehnout jen na mutaci, počet generací roste. Zajímavý je poznatek, že velká hodnota P_k přispívá k dobrým výsledkům u obvodů, kde je v pravdivostní tabulce určitá symetrie. Tabulka č.1 pak shrnuje výsledky nejlepších nalezených dvojic P_k a P_m a to včetně časových údajů na systému s taktovacím kmitočtem 90MHz. Počet nutných generací je u obou variant mutace téměř shodný, ale protože jednobitová mutace trvá výrazně kratší dobu, celkové časové údaje hovoří jednoznačně pro použití tohoto způsobu mutace.

Závěr

Uvedená hardwarová realizace SGA v obvodu FPGA spolu s rekonfigurovatelným obvodem tvoří samostatně se vyvíjející kombinační obvod, který je v rámci jistých omezení schopen měnit dynamicky svojí konfiguraci dle požadované výstupní odezvy. Testy prokázaly, že jednobitová mutace jedinců je srovnatelná s vícebitovou, avšak vyžaduje daleko méně systémových nároků. Při testech také velmi potěšil fakt, že evoluce si je schopná poradit i se simulovanými poruchami na rekonfigurovatelném obvodu, algoritmus se poruše přizpůsobil a vytvořil takový kombinační obvod kde se závada neprojevila, či kde jí dokonce využil. Tento úkaz jen potvrzuje analogii s biologií. SGA se přizpůsobuje, stejně jako živé organismy, svému okolí.

Literatura

- [1] Lažanský, Mařík, Štěpánková: *Umělá inteligence 3*. Academie, 2003. ISBN 80-200-0472-6.
- [2] Sekanina Lukáš, Mikušek Petr: *Analysis of Reconfigurable Logic Blocks for Evolvable Digital Architectures*, In: *Lecture Notes in Computer Science*, roč. 2008, č. 4974, DE, s. 144-153, ISSN 0302-9743.