IMPLEMENTATION OF AES ALGORITHM ON FPGA

David Smékal
Master Degree Programme (2), FEEC BUT
E-mail: xsmeka11@stud.feec.vutbr.cz

Supervised by: Jan Hajný
E-mail: hajny@feec.vutbr.cz

Abstract: This paper presents a VHDL (Very High Speed Integrated Circuit Hardware Description Language) implementation of 128-bit AES (Advanced Encryption Standard) on FPGA card (Field-Programmable Gate Array) using Virtex-7 FPGA chip manufactured by Xilinx company. In this project our main concern is to implement all modules of this algorithm on hardware.

Keywords: Cryptography, FPGA, AES, VHDL

1 ÚVOD

Článek se věnuje problematice zabezpečení vysokorychlostních komunikačních systémů. Je zde především představen vybraný algoritmus, který se dnes řadí mezi nejrozšířenější. Jedná se o šifrovací algoritmus AES. Cílem je implementace tohoto algoritmu na platformě síťových karet FPGA. Jde o programovatelná hradlová pole, která umožňují vývoj hardwarově akcelerovaných aplikací.

FPGA jsou programovatelné logické obvody, které lze naprogramovat až po jejich výrobě. V dnešní době obsahují především programovatelné logické bloky, RAM paměti a multiplexery. Místo vývoje zařízení pro konkrétní funkci, se zvolená funkce nastaví po výrobě a lze ji i průběžně měnit v závislosti na požadavcích konkrétního zadání. Výhodou těchto logických obvodů je jejich univerzálnost. Pro tvorbu uceleného projektu bylo zvoleno vývojové prostředí Vivado® od společnosti Xilinx, které umožňuje odsimulovat jednotlivé kroky algoritmu.

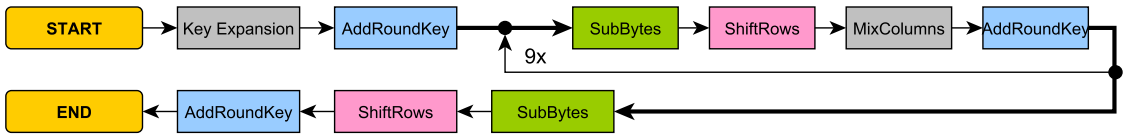
Proč se vlastně šifrování implementuje na hardware? Vždyť je přece spousta softwarových mechanizmů na zabezpečení dat. Implementace na FPGA je mnohem rychlejší a efektivnější, než softwarová, a to řádově až v desítkách Gbps.

2 POPIS ŠIFROVÁNÍ AES

Advanced Encryption Standard, neboli AES, je algoritmus používaný k šifrování dat. Jedná se o symetrickou blokovou šifru, která zpracovává data rozdělená do bloků pevně dané délky 128 bitů. Tyto bity jsou uspořádány do matice 4×4, kdy jedna buňka matice odpovídá jednomu bajtu. K šifrování a dešifrování se používá stejný klíč o velikosti 128, 192 nebo 256 bitů. V tomto článku se používá klíč 128 bitů, který je předem vypočítán. Algoritmus lze rozdělit na tři části – Iniciační část (Key Expansion, AddRoundKey), iterace (SubBytes, ShiftRows, MixColumns, AddRoundKey) a závěrečná část (SubBytes, ShiftRows a AddRoundKey) [1].

Na začátku šifrování se provede expanze klíče. Ke stavové matici 4×4 vytvořené z bloku 128bitů (8×16 bajtů) se v šifře přičte klíč. Poté se devětkrát provede hlavní šifra, což se běžně označuje jako runda. Počet provedení rundy se odvíjí v závislosti na použití délky klíče. Počet 9 odpovídá klíči 128 bitů. Každá runda se skládá ze substituce bajtů stavové matice (SubBytes), rotace řádků (ShiftRows), následně substituce sloupců (MixColumns). Na konci každé rundy se k matici přičte rundovní (iterační) klíč (AddRoundKey). V závěrečné části se opět uskuteční substituce bajtů, rotace

řádků a poslední fází je přičtení klíče finální rundy. Ve výsledné matici jsou uloženy bajty šifrového textu. Tyto popsané kroky jsou zobrazeny ve vývojovém diagramu 1.

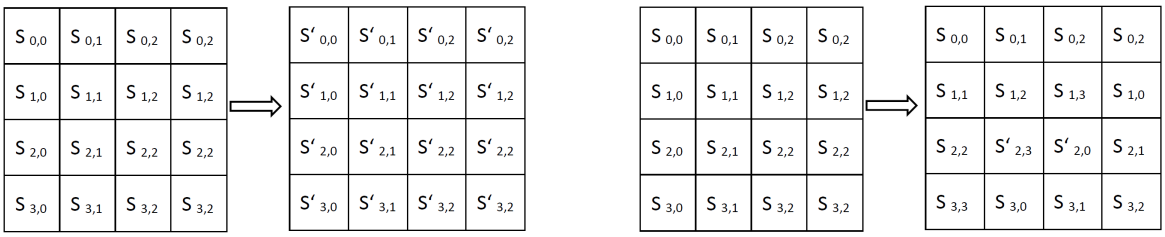


Obrázek 1: Jednotlivé kroky šifrování AES

2.1 SubBytes – substituce bajtů

Jedná se o prostou substituci, kde každému vstupnímu bajtu je přiřazena předem daná hodnota výstupního bajtu. Přiřazení se provádí dle známé tabulky. Každý bajt se rozdělí na dvě hexadecimální číslice. Pomocí první se určí řádek a pomocí druhé sloupec v tabulce. V dané buňce se pak přečte substituovaný bajt.

Vstupní blok dat, tedy 16 bajtů, rozdělíme na jednotlivé bajty, se kterými následně pracujeme. Lépe řečeno porovnáváme aktuální bajt se substituční tabulkou, kterou máme definovanou jako výběrové přiřazení. Deklarace přiřazení se skládá ze všech 256 možných kombinací vstupního bajtu. Pakliže vstupní blok dat je 16 bajtů, každý bajt se nahradí novou hodnotou určenou ze substituční tabulky. Operace znázorněna na obrázku 2.



Obrázek 2: Aplikace SubBytes (vlevo) a transformace ShiftRows(vpravo)

2.2 ShiftRows – rotace řádků

Při rotaci řádků se upravují jednotlivé řádky matice následujícím způsobem. V prvním řádku matice se neprovede řádná změna. Ve druhém řádku se provede rotace vlevo o jeden bajt, ve třetím řádku rotace vlevo o dva bajty a ve čtvrtém řádku se provede rotace vlevo o tři bajty.

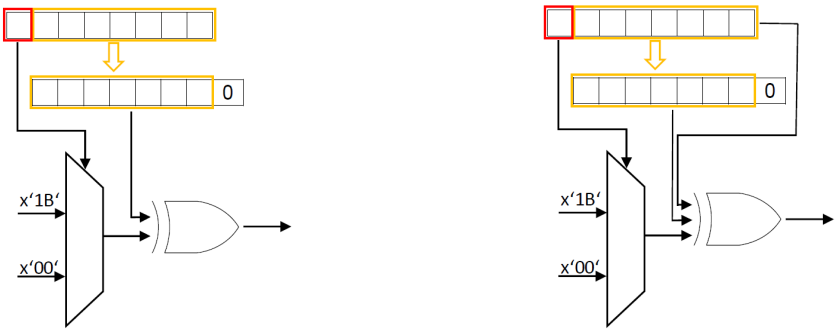
Je důležité si uvědomit, že matici reprezentujeme vektorem. Je třeba mít na paměti, že první 4 bajty tvoří první řádek, další 4 bajty druhý řádek atd. Pracujeme proto s indexem jednotlivých bitů, které transformujeme dle teoretických pravidel. Transformace znázorněna na obrázku 2.

2.3 MixColumns – substituce sloupců

Transformace MixColumns vychází z vynásobení získané matice tzv. mixovací maticí viz tabulka č.1. Sloupec původní matice se násobí mixovací maticí, a vznikne tak sloupec nové transformované matice. Násobení jedničkou znamená ponechání původní hodnoty. Násobení dvojkou znamená posuv původní hodnoty o jeden bit vlevo. Násobení trojkou znamená posuv původní hodnoty o jeden bit vlevo a následné sečtení (XOR) s původní hodnotou. Pokud je výsledek vyšší než 255, pak se k němu ještě přičte hodnota 1B v hexadecimálním tvaru. Operace násobení dvěma a násobení třemi jsou zřejmé z obrázku č.3. Pracuje se s jedním bajtem, vykoná se posuv a následné sečtení.

2	3	1	1
1	2	3	1
1	1	2	3
3	1	1	2

Tabulka 1: Mixovací matice



Obrázek 3: Násobení dvěma (vlevo) a třemi (vpravo)

2.4 AddRoundKey – přičtení rundovního klíče

Přičtení rundovního klíče je poslední transformací. Odvození dílčích klíčů o stejném formátu jako původní matice, tj. 4×4, jsme provedli softwarově a klíče se uloží do paměti nebo registrů na čipu. Následná operace sečtení matice a klíče je prováděna pomocí exklusivního logického součtu XOR.

Vstupní blok dat je stejně velký jako šifrovací klíč, tedy 128 bitů. Obě tyto hodnoty jsou známé, takže nezbývá než provést logickou operaci XOR.

3 REALIZACE

Při praktické realizaci návrhu zabezpečení využíváme síťovou FPGA kartu COMBO-80G od společnosti Invea-Tech, umožňující vývoj hardwarově akcelerovaných aplikací. Karta je osazena výkonným FPGA čipem Virtex–7 firmy Xilinx. Podporuje technologie 40G a 10G Ethernet. Využívá sběrnici PCI Express pro vysokorychlostní přenosy dat mezi kartou a hostitelským počítačem.

Za pomoci vytvořených testbenchů pro jednotlivé komponenty se odsimulovaly výše uvedené kroky algoritmu. Následně se sjednotily všechny komponenty do jednoho projektu Vivada a provedla se finální simulace. Ta představuje implementaci a ověření výsledků celého procesu šifrování.

4 ZÁVĚR

Pomocí simulace jsme ověřili, že data byla správně zašifrována. Vstupní data prošla přes 4 nezávislé bloky, kdy jsme na výstupu dostali zašifrovaný blok dat. Po ověření správnosti algoritmu se přistoupilo k syntéze, která umožní tvorbu firmwaru pro konkrétní hardwarové zařízení a funkčnost bude možno otestovat v reálné síti.

REFERENCE

[1] BURDA, Karel. *Aplikovaná kryptografie*. 1. vyd. Praha: VUTIUM, 2013, 255 s. ISBN 978-80-214-4612-0.

[2] PINKER, Jiří, POUPA, Martin. *Číslicové systémy a jazyk VHDL*. 1. vyd. Praha: BEN - technická literatura, 2006, 349 s. ISBN 80-730-0198-5.