



POLYTECHNIQUE Montréal

TECHNOLOGICAL

Presented by John Doe March 26, 2024

IP Core Identification in FPGA Configuration Files using Machine Learning Techniques

Mahmood et al.

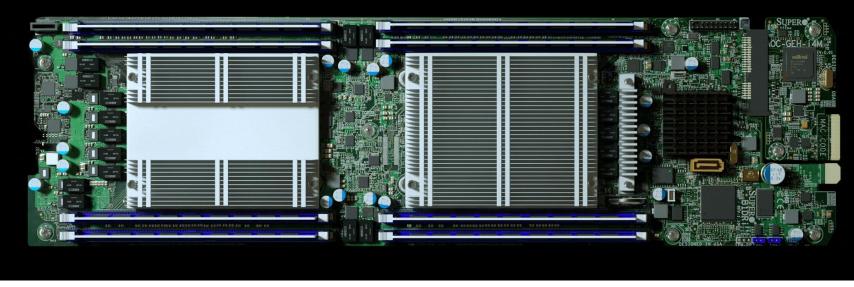


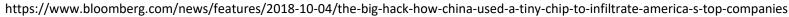
US Edition ∨

Businessweek | Feature

The Big Hack: How China Used a Tiny Chip to Infiltrate U.S. Companies

The attack by Chinese spies reached almost 30 U.S. companies, including Amazon and Apple, by compromising America's technology supply chain, according to extensive interviews with government and corporate sources.







J. 7

US Edition ∨

Businessweek | Feature

The Big Hack: How China

Infiltrate U.S. Companies

The attack by Chinese spies reached almost 30 U.S.

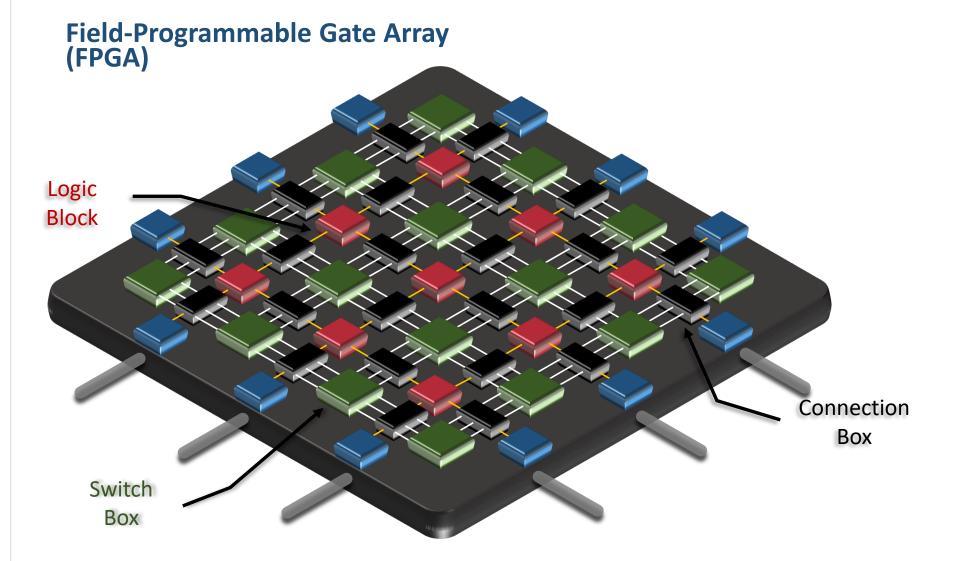
Used a Tiny Chip to

Bloomberg Businessweek The Big Hack How China used a tiny chip to infiltrate America's top companies

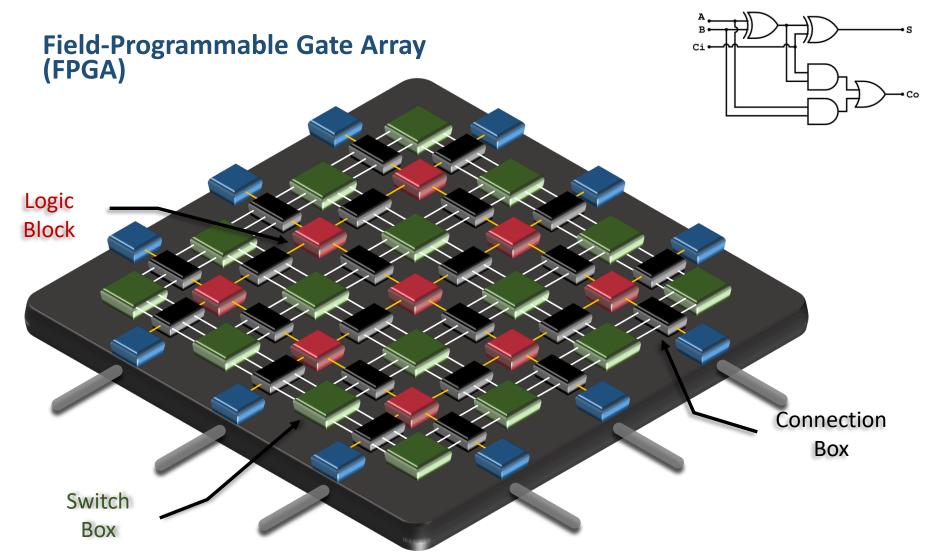
companies, including Amazon and Apple, by compromising America's technology supply chain, according to extensive interviews with government and corporate sources.

https://www.bloomberg.com/news/features/2018-10-04/the-big-hack-how-china-used-a-tiny-chip-to-infiltrate-america-s-top-companies

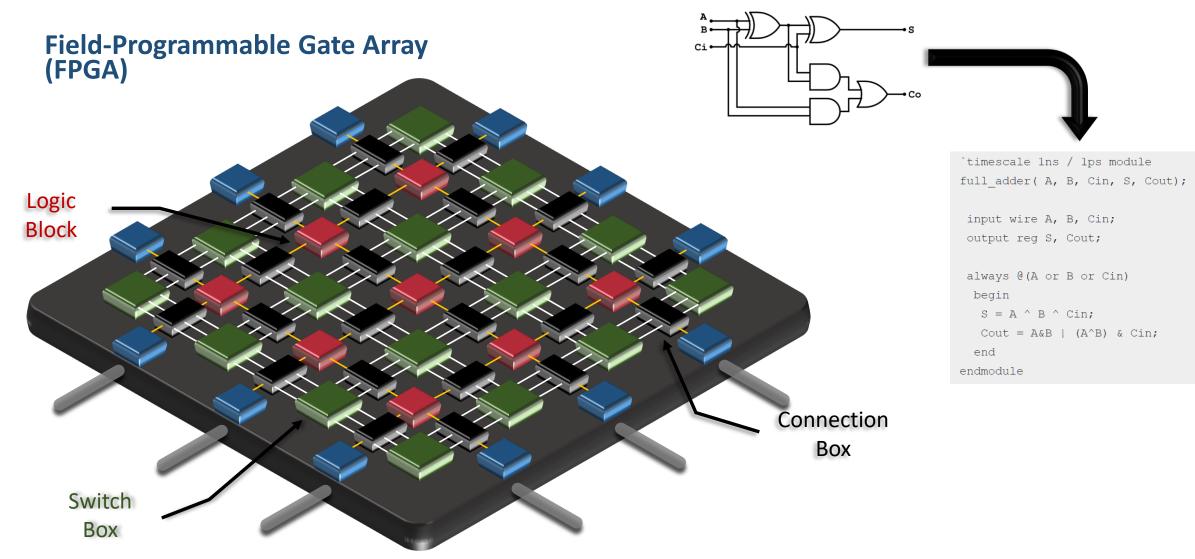




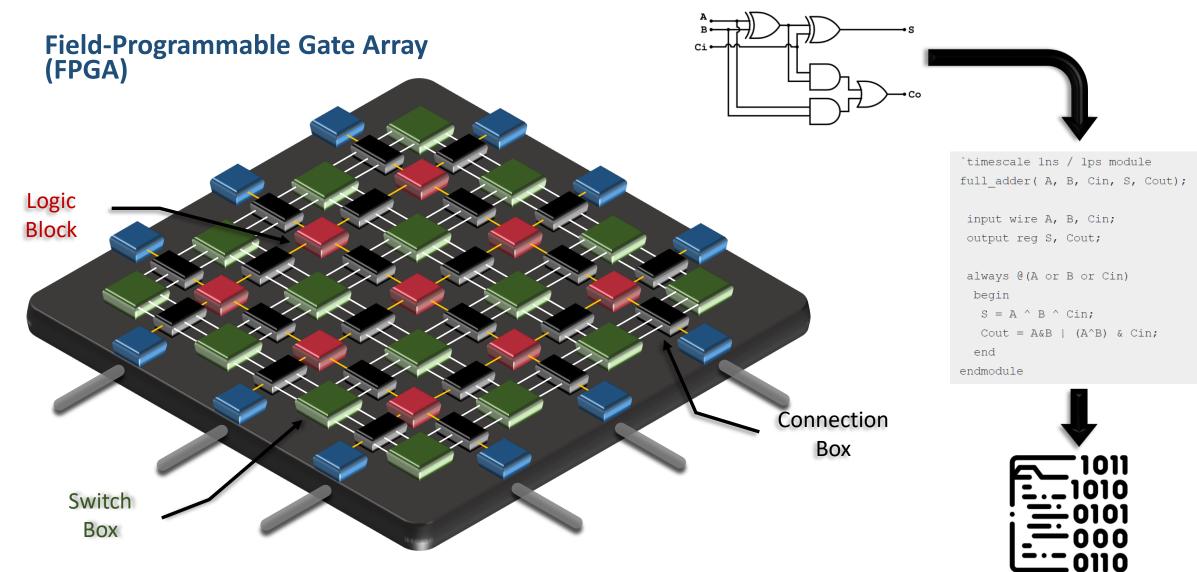




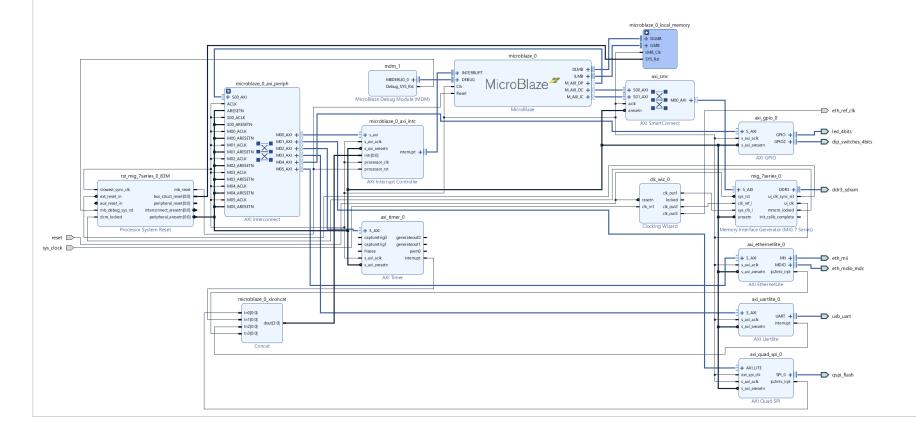












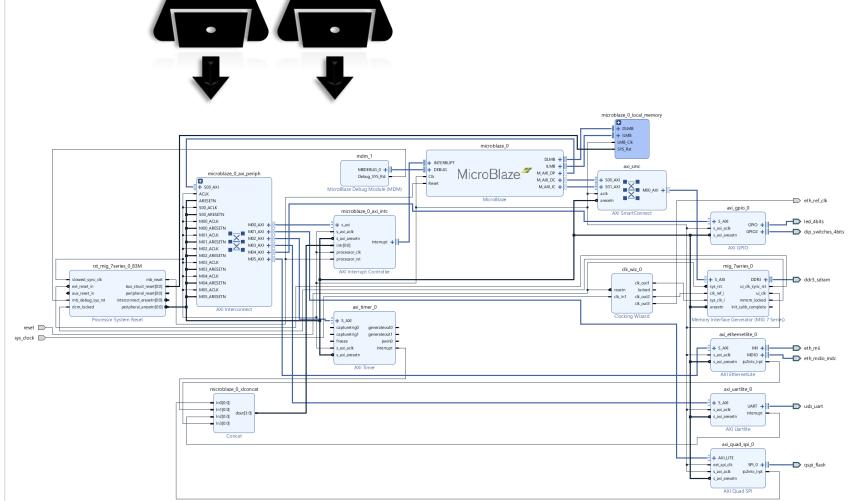




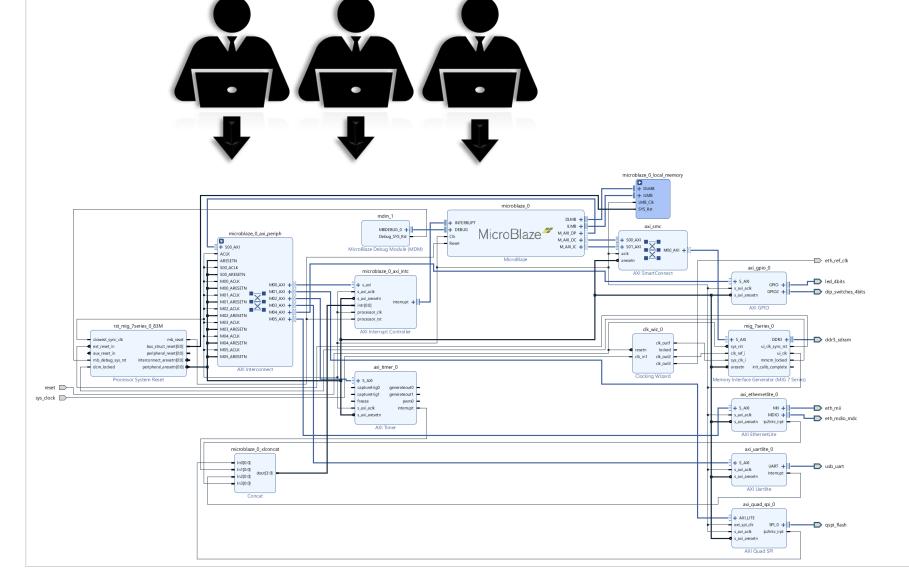
microblaze 0 local memory . + DLMB + ILMB LMB CIk microblaze_0 SYS_Rst mdm 1 DLMB + + INTERRUPT MicroBlaze axi smo MBDEBUG_0 + + DEBUG microblaze_0_axi_periph Clk Debug_SYS_Rst + S00_AXI + SOO_AXI ACLK + S01_AXI aclk aresetn M_AXI_IC + MicroBlaze Debug Module (MDM) ACLK ARESETN S00_ACLK S00_ARESETN M00_ACLK MicroBlaze - eth_ref_clk axi_gpio_0 microblaze_0_axi_intc led_4bits + S AXI M00_AXI + + s_axi GPIO + s_axi_ack M00_ARESETN M00_ARESETN M01_ACLK M01_ACLK M01_ARESETN M01_AXI + M02_AXI + M02_AXI + M02_AXI + M02_AXI + s_axi_aclk GPIO2 + — dip_switches_4bits s_axi_aresetn s_axi_aresetn interrupt + M01_ACLK AXI GPIO processor_clk M02_ARESETN M03_ACLK M03_ARESETN M04_ACLK M05_AXI + processor_rst mig_7series_0 rst_mig_7series_0_83M clk_wiz_0 AXI Interrupt Controller slowest_sync_clk mb_reset -+ S_AXI DDR3 + ddr3_sdram M04_ARESETN clk_out1 ext reset in bus struct reset[0:0] sys_rst ui_clk_sync_rst M05_ACLK locked resetn aux reset in peripheral reset[0:0] _____ clk ref i ui_ck 🗕 M05_ARESETN clk_in1 clk_out2 -- mb_debug_sys_rst interconnect_aresetn[0:0] sys_clk_i mmcm_locked clk_out3 🗕 dcm_locked aresetn init_calib_complete peripheral_aresetn[0:0] 🔶 axi timer 0 + S_AXI - capturetrig0 reset 📄 generateo capturetrig1 generateout1 freeze pwm0 axi_ethemetlite_0 sys_clock D s_avi_ack MDIO + Ch_mdio_mdc s_axi_aclk interrupt s_axi_aresetn s_axi_aresetn ip2intc_irpt AXI Timer AXI EthernetLite microblaze_0_xlconcat axi_uartlite_0 - In0(0:0) + S_AXI -D usb_uart In1[0:0] dout[3:0] UART + s_axi_aclk interrupt In2[0:0] s_axi_aresetn In3[0:0] AXI Uartlite axi_quad_spi_0 + AXI_LITE ext_spi_clk SPI_0 + -D qspi_flash s_axi_aclk ip2intc_irpt s_axi_aresetn AXI Quad SPI





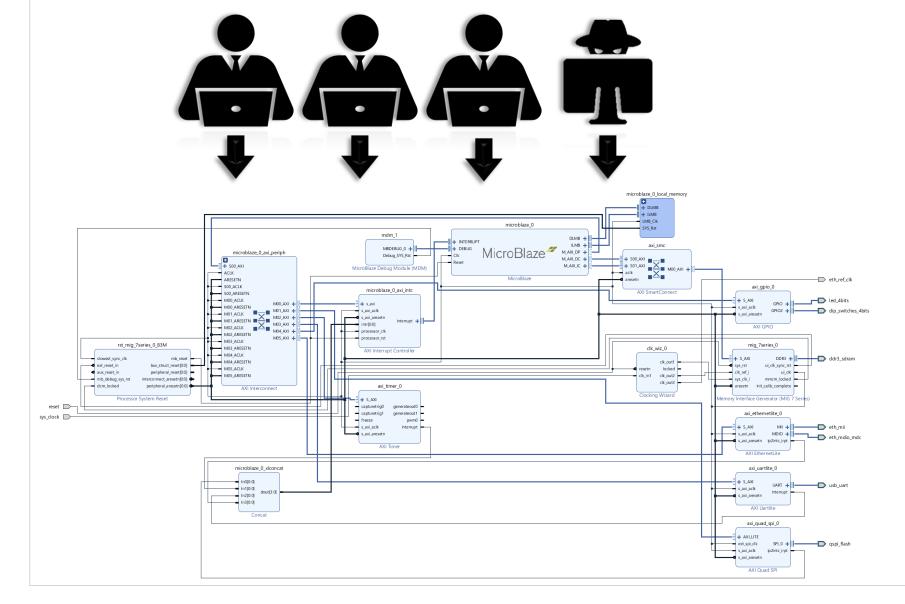






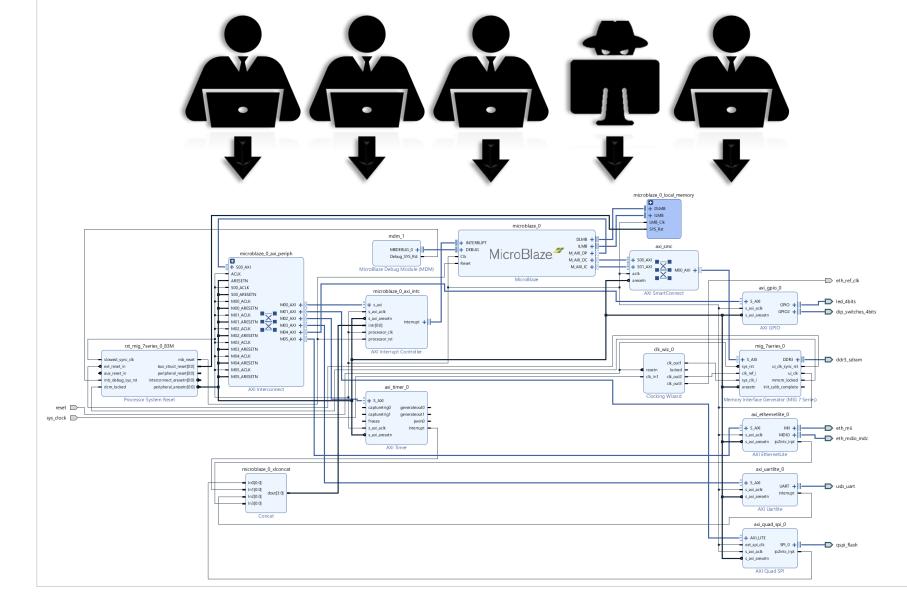










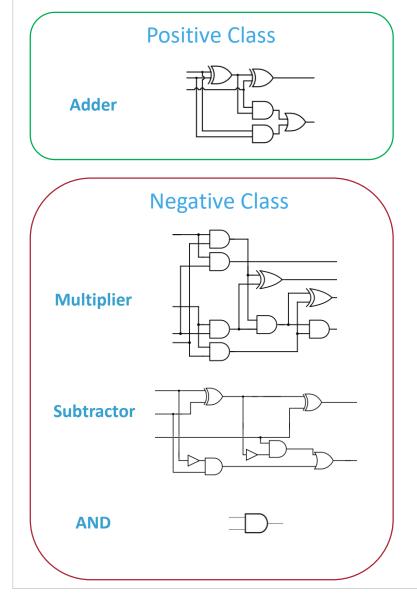






One-vs-All Classification:

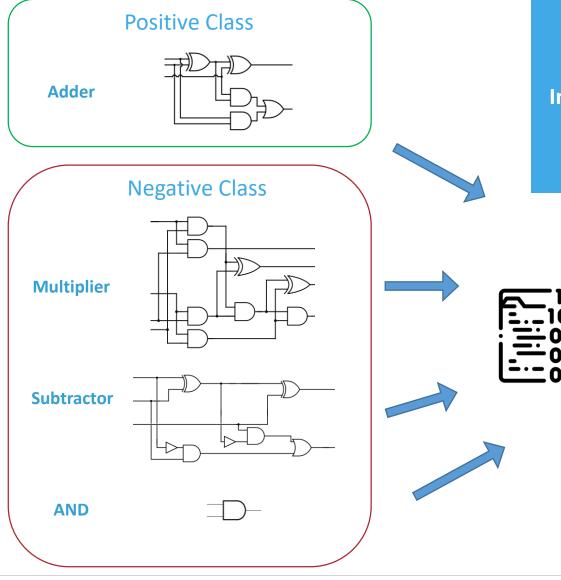
Individual Neural network models are trained to identify one hardware module (eg. Adder) against all other modules.



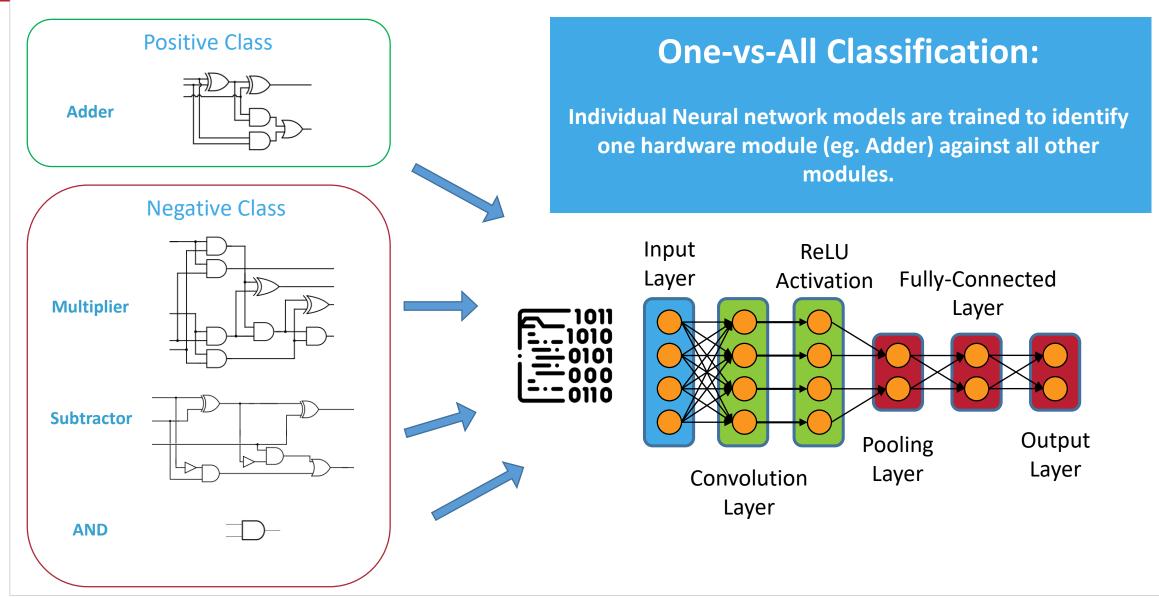


One-vs-All Classification:

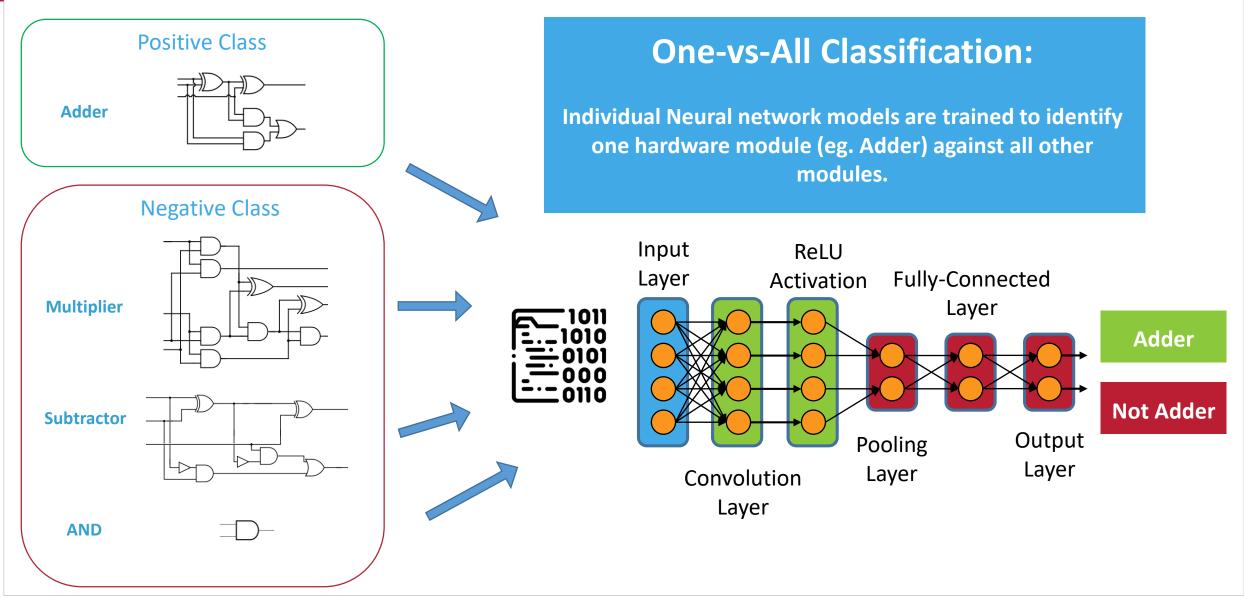
Individual Neural network models are trained to identify one hardware module (eg. Adder) against all other modules.



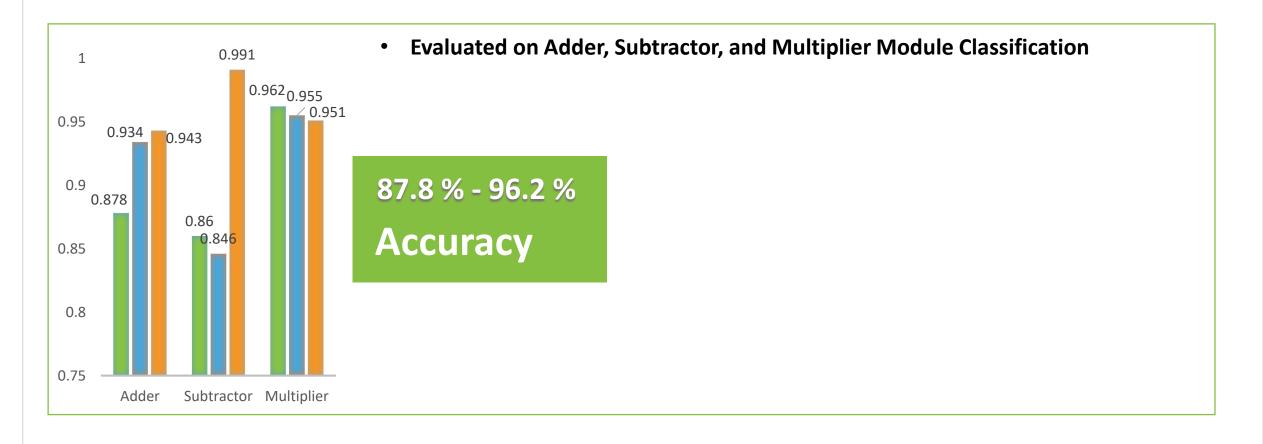




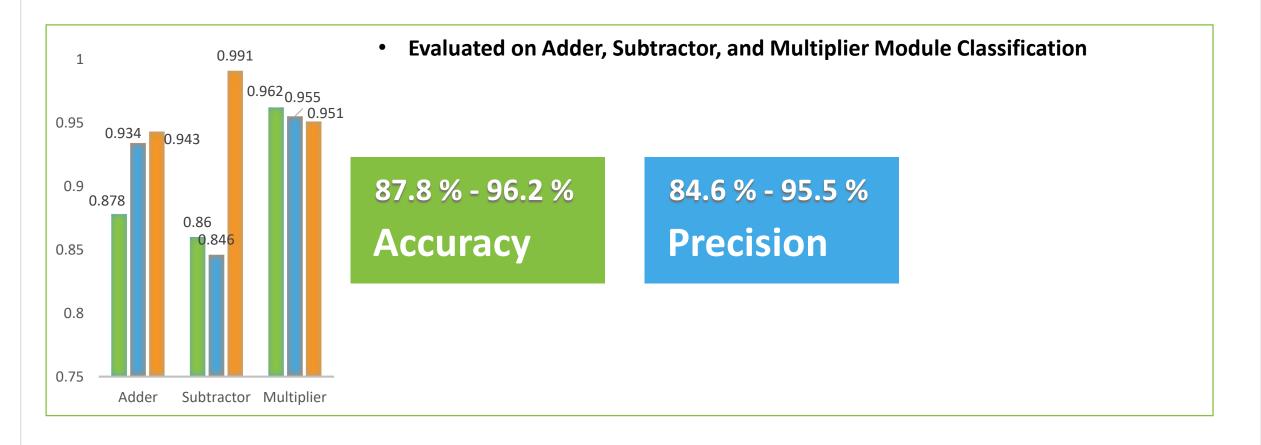




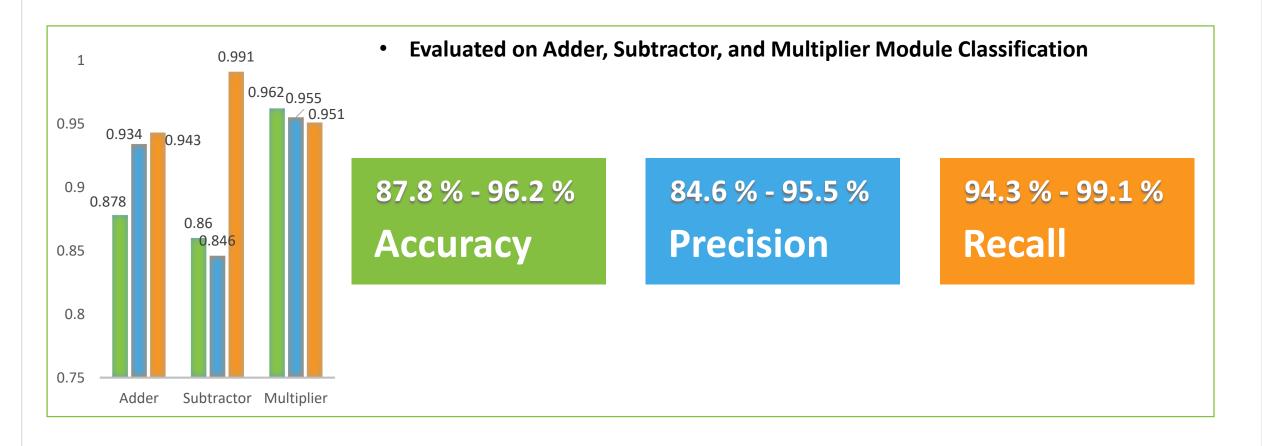




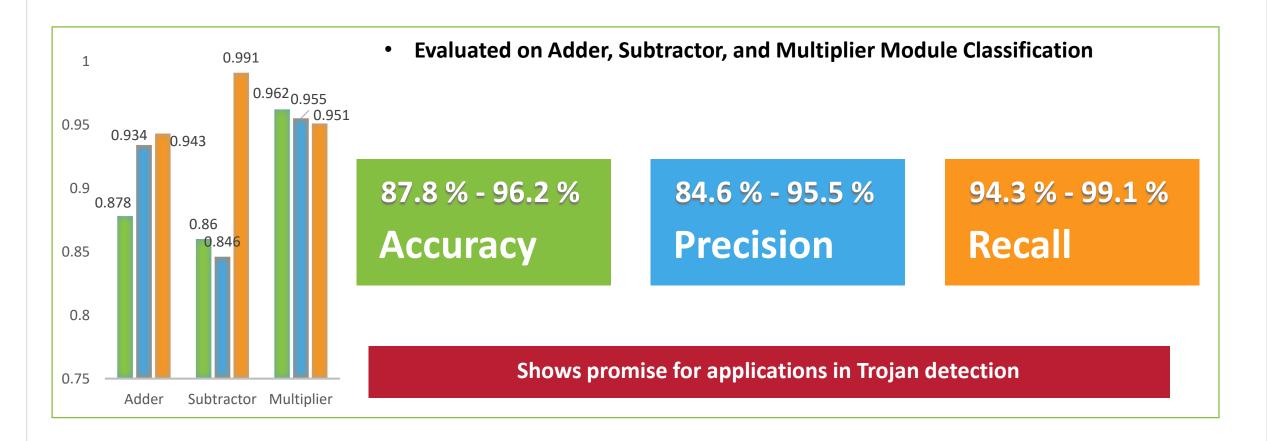














Conclusion

Paper Presentation: IP Core Identification in FPGA Configuration Files using Machine Learning Techniques

There is a Need for Post-Development Integrity Verification in FPGA Designs

Main Findings

 CNNs can successfully identify know logic blocks in generated binaries with an accuracy rate of over 87.8 %, high precision and high recall



Conclusion

Paper Presentation: IP Core Identification in FPGA Configuration Files using Machine Learning Techniques

There is a Need for Post-Development Integrity Verification in FPGA Designs

Main Findings

 CNNs can successfully identify know logic blocks in generated binaries with an accuracy rate of over 87.8 %, high precision and high recall

Implications

Enable developers to scan their designs for known malicious logic



Conclusion

Paper Presentation: IP Core Identification in FPGA Configuration Files using Machine Learning Techniques

There is a Need for Post-Development Integrity Verification in FPGA Designs

Main Findings

 CNNs can successfully identify know logic blocks in generated binaries with an accuracy rate of over 87.8 %, high precision and high recall

Implications

Enable developers to scan their designs for known malicious logic

Future Works

• Test the approach with complex logic blocks



john.doe@polymtl.ca



POLYTECHNIQUE Montréal

TECHNOLOGICAL ∃ UNIVERSITY

