

Burhanuddin Qadir

☎ (416)-618-4101 | ✉ Burhanqb@hotmail.com | [in burhanuddinqadir](https://www.linkedin.com/in/burhanuddinqadir) | [github burhanqb](https://github.com/burhanqb)

EDUCATION

McMaster University

Bachelor in Computer Engineering and Business Management 3.7/4 CGPA - Deans List

Sept. 2019 – Apr. 2026

Hamilton, ON

WORK EXPERIENCE

Design Verification Engineer

Advanced Micro Devices (AMD)

May. 2023 – Aug. 2024

Markham, ON

- Developed & debugged performance counters using **Verilog** & **Verdi** for Dual SDP, created technical design docs
- Managed Design Checks (**Lint**, **CDC**, **Diffbia**, **Vaelab**) on 8+ projects with 5+ variants & maintained a web app
- Developed an automated memory generation script using **Verilog** & **Perl (VPP)** enabling all types of memories (P-type, Q-type, L-type) to be generated based on user specifications, increasing the teams efficiency by 10%
- Worked in an **Agile** team using **JIRA** & **Confluence**, created & managed 10+ **Jenkins** pipelines using **Groovy**, **Python** & **Github**, added Teams/Email notifications, 250+ live test outputs enhancing team performance by 25%
- Enhanced **Perl** & **Verilog (VPP)** scripts using **VIM** & **Perforce**, reducing processing time by over 10 hours
- Presented to 500+ new AMD CO-OP's & mentored peers, presented to 50+ McMaster students at industry night

Google Developer Student Club – Web Developer

McMaster University

Sept. 2022 – Present

Hamilton, ON

- Developed online GDSC merchandise store using **Git** & **Python** on **Vs Studio** & maintained GDSC website

Intermediate Software Engineer

The Canadian Imperial Bank of Commerce (CIBC)

Sept. 2021 – Apr. 2022

Toronto, ON

- Worked in **Agile** team using **JIRA** & **Confluence**, developed 15 new REST API endpoints using **SpringBoot** & **MS SQL**, validated using **Postman** for Goal Planning application installed in all the CIBC branches
- Reduced infrastructure cost by over 50% by designing & implementing **docker** & container based technologies using **REH OpenShift** environment with **OCP CLI**
- Deployed 15 API to DEV,UAT,PROD env by creating **Jenkins** pipelines integrated with **Github**, **JFrog Artifactory**, **SonarQube** code analysis etc.
- Maintained 40 microservices repos in **GitHub** & 10 **JFROG Artifactory** repos for storing & promoting build artifacts & **Docker** images

ROS Development and Robotics

CrossWing Inc.

Jan. 2019 – Jun. 2019

Aurora, ON

- Developed 2 camera modules for the autonomous robot created 100% accurate 3D models using **Rviz** & **ROS**
- Mapping localization of the robot covering 100% of the area with **RTabmap SLAM**, also created Virtual 3D simulation environment using **Gazebo**, used terminal commands automated the camera operations on 2 robots

TECHNICAL SKILLS

Logic Design: Quartus II, Synopsys Verdi, Verilog HDL, Block Diagram/Schematic, Waveform Simulation, RTL

Languages: Python, Java, C, Bash, Shell script, JSON, HTML, Java Script, Assembly, Perl, VPP, Ruby

Developer Tools: Github, Jenkins, Docker, VS Code, PyCharm, Eclipse, JFROG Artifactory, SonarQube, Jira, VIM

Developer Tools (cont'd): Confluence, Perforce, Github Co-pilot, Internal ChatGPT, GVIM

Robotics: ROS, Rviz, RTabmap, SLAM, Gazebo, Intel NUC, Intel RealSense Camera

Database: R language, Data Structures, Algorithms, Hashing, Microsoft SQL, Matlab

PROJECTS

Logic Design | *Quartus II, Verilog HDL, Block Diagram/Schematic, Waveform Simulation*

- Designed a Datapath controller, given the functional requirements created a **state diagram**, **state table**, **truth table**, **karnaugh maps**, lastly a **schematic model**. Ran functional simulations using **Waveform**
- Created 4-bit counter using **Quartus II block diagram/schematic** & created a 3-8 decoder using **Verilog HDL**
- Created a full adder that took two binary inputs displayed a decimal value onto a 7-segment display

Pacemaker | *Github, PyCharm, Vs Studio, Tkinter*

- Developed a GUI for a pacemaker using **Tkinter** for doctors to monitor patient pacemaker levels
- Implemented **PostgreSQL** database to store up to 10 users, each user had a username password with their unique pacemaker levels which could be saved/edited by using **matplotlib** the values are graphically viewable
- Collaborated with **Simulink Stateflow** to create serial communication between the GUI the pacemaker