|  |
| --- |
| **عنوان پروژه: طراحی زبان برنامه نویسی سطح بالا برای طراحی الگوریتم های کوانتومی** |
| **واژگان کلیدی**: Quantum Programming، Quantum Compiler، Quantum Algorithm Design |
| **کاربردها**: طراحی الگوریتم های کوانتومی و توابع محاسباتی برای فرایند های محاسباتی صنایع مختلف (شیمی، مالی، بیولوژی، ...) توسط یک کاربر بدون دانش تخصصی از محاسبات کوانتومی |
| **شرایط موجود در کشور**: تابحال چنین پروژه ای در کشور انجام نشده است، اما استعداد ها، پتانسیل ها و نیرو های لازم برای پیاده سازی چنین پروژه ای در کشور فراهم است. جهت پیاده سازی این پروژه به مهندسین نرم افزار و متخصصین در زمینه طراحی الگوریتم و آشنا با الگوریتم های کوانتومی موردنیاز است تا بتوانند این زبان برنامه نویسی را پیاده سازی نمایند. خوشبختانه مهندسین نرم افزار و طراح های الگوریتم ماهری در کشور برای اینکار وجود دارند. |
| **شرایط موجود در دنیا:**  شرکت ها و دانشگاه های بسیاری با ارائه راه حل های طراحی الگوریتم و زبان های برنامه نویسی کوانتومی سطح بالا بر روی این پروژه فعالیت کرده اند. از جمله زبان های برنامه نویسی می‌توان به محصول شرکت Classiq Technologies، زبان SilQ و نرم افزار Orquestra شرکت Zapata Computing اشاره نمود. این شرکت ها همگی توسط نیرو های جوان و تازه کار اما متخصص در این زمینه ایجاد شده اند و از طول عمر آنها زمان زیادی نمیگذرد. طبق گزارش های اینترنتی، درامد حاصل از فروش راه حل های طراحی الگوریتم های کوانتومی توسط شرکت Zapata Computing معادل 16.2 میلیون دلار در سال بوده و این رقم برای شرکت نوپای Classiq Technologies (متعلق به رژیم صهیونیستی) معادل 2.8 میلیون دلار در سال است. |
| **بیان و تشریح مسئله:**  طراحی الگوریتم کوانتومی (Quantum Algorithm Design) نسخه کوانتومی طراحی به کمک کامپیوتر (Computer Aided Design) است. هدف QAD حل همان مشکلی است که سال‌ها پیش توسط CAD الکترونیکی حل شد: با بزرگ‌تر شدن و پیچیده‌تر شدن مدارها، طراحی آنها با دست غیرممکن می‌شود. در حالی که یک مهندس الکترونیک بدون شک می تواند یک مدار شامل 20 گیت منطقی طراحی کند، اما آی سی های الکترونیکی جدیدتر امروزی میلیون ها گیت دارند. ایجاد Netlist برای این تراشه ها با دست غیرممکن است. با این حال، ارائه یک مدل سطح بالا برای یک رایانه و درخواست از نرم افزار برای تبدیل آن به یک مدار ممکن است، از این رو پلتفرم‌های CAD محبوبیت دارند. همانطور که یک طراح وب نیازی به درک نحوه عملکرد یک گیت CMOS ندارد، یک مهندس نرم افزار کوانتومی نیز نباید نیازی به درک پیچیدگی های سخت افزار کوانتومی سطح پایین داشته باشد.  طراحی الگوریتم کوانتومی به طراح اجازه می دهد تا به جای اجرای سطح پایین، روی عملکرد الگوریتم تمرکز کند. لذا برای طراحی توابع و الگوریتم های کوانتومی با تعداد دلخواه کیوبیت نیز لازم است یک مدل سطح بالا فراهم شود تا بتواند الگوریتم های مختلف کوانتومی و توابع سطح بالا را بصورت تابع از ورودی دریافت نموده و مدار کوانتومی آن را بطور خودکار سنتز نموده و خروجی آن بصورت یک کد برنامه نویسی زبان سطح پائین کوانتومی مانند Qiskit، Q# و یا CirQ باشد. درنتیجه، استفاده از مدل های QAD در طراحی الگوریتم ها و حل مسائل در کامپیوتر های کوانتومی امری ضروری محسوب می‌شود و می‌تواند به طراحی در سطح سیستم کمک فراوانی کند. عرض و عمق مدار، استفاده از یک مجموعه گیت خاص در طراحی، دقت موردنظر و نحوه اتصال کیوبیت ها می‌تواند از جمله محدودیت هایی باشد که در طراحی یک الگوریتم موردنظر توسط کاربر تعیین می‌شوند تا خروجی مورد نظر را در زمان معقول ارائه دهد.  با مدل های QAD میتوان کد خود را به سرعت بین پلتفرم های سخت افزاری مختلف تبدیل کنید، چرا که مشخص نیست که کدام پلتفرم برای پیاده‌سازی کامپیوتر یونیورسال برنده خواهد شد. مزیت دیگر این است که برنامه نویسی سطح پایین به سرعت با پیچیدگی کامپیوترهای کوانتومی غیرممکن می شود و دیباگ کردن کد سطح بالا بسیار آسان‌تر از معادل زبان های اسمبلی کوانتومی است.  فاز های پیاده سازی پروژه:   1. ماژولار کردن الگوریتم های کوانتومی آکادمیک (دویچ جوزا – سایمون – برنیشتین وزیرانی – گروور – شور – HHL و ...) 2. ماژولار کردن مسائل صنعتی در مقیاس کوچک – متوسط – بزرگ (امکان سنجی اولیه موردنیاز است.) 3. توسعه دادن نرم افزار برای کاربرد ها و الگوریتم های مختلف |
| **اهداف پروژه:** فراهم نمودن موتور سنتز طراحی الگوریتم های مختلف برای تعداد کیوبیت های دلخواه. هدف اصلی این پروژه فراهم نمودن راه حل های ماژولار و توابعی جهت تسهیل فرایند طراحی الگوریتم است. البته شاید برای تمامی مسائل امکان طراحی عمومی وجود نداشته باشد، اما تمامی توابع تهیه شده الگوریتم ها باید از کاربر آرگومان های کلی و قید های اصلی مسئله را دریافت کنند و بتوانند مدار کوانتومی لازم را تولید کرده و خروجی های آن را به کاربر تحویل بدهد. |
| **مرحله‌ی بعدی پروژه**: این پروژه بصورت توسعه ای بوده و میتواند دائما مورد توسعه و پیشرفت قرار گیرد. در مراحل اولیه میتوان الگوریتم های پرکاربرد و مورداستفاده مکرر را ماژولار نمود. در مراحل بعدی میتوان راه حل های الگوریتمی برای صنایع و مسائل مختلف تعریف و طراحی کرد. |
| **خروجی‌های مورد انتظار**: یک زبان برنامه نویسی و موتور سنتز الگوریتم های کوانتومی که بتواند قید ها، ورودی ها و محدودیت های مسئله را از کاربر دریافت کند و خروجی های لازم، مدار کوانتومی را با دقت تعیین شده به کاربر تحویل دهد.  برای اینکار لازم است ابتدا ابزار های مدلسازی الگوریتم در سطح تابع تعریف شود: مثلا امکان طراحی بلوک های تعریف شده مانند VQE، QAOA، Build Hermitian Matrix، Amplitude Estimation، Ansatz Preparation، Quantum Arithmatic و Combinatorial Optimization فراهم شده باشد و امکان فراخوانی آنها برای کاربر موجود باشد.  سپس ابزار های آنالیز مسئله باید فراهم شده باشند. یعنی میزان منابع لازم (تعداد کیوبیت، تعداد گیت، میزان درهمتنیدگی و سایر محدودیت های مسئله) در نظر گرفته شوند و بهترین سخت افزار برای نوع مسئله خاص پیشنهاد داده شود.  و درنهایت ابزار های سنتز باید طراحی شوند تا تعاریف و ورودی های تعریف شده را به طراحی های درسطح گیت تبدیل کنند تا در کامپایلر های متعارف قابل اجرا باشند. |