ظهور معماری پردازنده های گرافیکی همه منظوره به همراه مدل های برنامه نویسی CUDA و OpenCLبستر مناسبی را فراهم نموده است، تا برنامه های مختلف با هزینه ای اندک و توان مصرفی پایین بتوانند با سطح بالاتری از موازی سازی نخ های پردازشی اجرا شوند. پردازنده های گرافیکی همه منظوره امروزی با اجرای همزمان ده ها تا هزاران نخ پردازشی به صورت بسیار چشمگیری در حال ارائه ی عملکرد خیره کننده ای در پردازش داده های عظیم محاسبات هستند. در این مدل که واحد های پردازشی به صورت گسترده در حال اجرای چندین نخ پردازشی همزمان هستند، تاخیر دسترسی به حافظه ی اصلی که فرایندی با تاخیر بسیار بالا است، تا حدی مخفی خواهد ماند. متاسفانه پردازنده های گرافیکی برای اجرای بسیاری از برنامه ها به دلیل محدودیت های منابع مانند دسترسی به حافظه قادر نیست تمام کارایی خود را برای اجرای ان برنامه بکار گیرید، و این امر موجب از بین رفتن فرصت ها برای رسیدن به کارایی نهایی ان ها شده است. و یکی از عمده علت های این مساله نبود روی کرد مناسب در رویارویی با تاخیر دسترسی به حافظه می باشد. پردازنده های گرافیکی ذاتا به دلیل وجود تعداد زیادی از نخ های پردازشی در حال اجرا ، قابلیت ان را دارند تا با تعویض نخ های در انتظار حافظه با نخ های پردازشی اماده، تاخیر حاصل از دسترسی به حافظه را برای کل مجموعه مخفی نگاه دارند. اما این ویژگی تا حدی می تواند تاخیر را از کل مجموعه مخفی بدارد که نخ های پردازشی اماده برای اجرا وجود داشته باشد در غیر این صورت به دلیل حجمه ی بالای درخواست های حافظه واحد پردازش نیز در حالت بیکار و انتظار برای حافظه خواهد ماند و بدین شکل تاخیر حافظه غیر قابل پوشش خواهد شد و بر کارایی کل مجموعه تاثیر سوء خواهد گذاشت.

ما در این پایان نامه قصد داریم تا قابلیت تاخیر پذیری پردازنده های گرافیکی را بهبود ببخشیم تا در مواجهه با دستورالعمل

عظیم محاسباتی و پردازشی همراه با درخواست های حافظه ای بیشتر، بتوانیم کارایی پردازنده های همه منظوره ی گرافیکی را با کاهش زمان بیکاری هسته های پردازشی حفظ نماییم. در روش پیشنهادی ایده بر سر ان است تا با اعمال اولویت به درخواست های مختلف حافظه ای ،به ان واحد پردازشی که احتمال وقوع حالت بیکاری در ان از همه بالاتر است میزان زمان بیکاری هسته ی پردازشی بحرانی را کاهش دهیم تا بدین شکل کارایی کل مجموعه را حفظ نماییم.