**Technion – Israel Institute of Technology**

**The Andrew and Erna Viterbi Faculty of Electrical & Computer Engineering**

**VLSI Laboratory**

**Project A (044167)**

**Semester Spring 2023**

**Hardware Accelerator for Genetic Local Sequence Alignment**

**![תמונה שמכילה דפוס, צבעוני, מלבן, בד

התיאור נוצר באופן אוטומטי](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAYABgAAD/4RDuRXhpZgAATU0AKgAAAAgABAE7AAIAAAAMAAAISodpAAQAAAABAAAIVpydAAEAAAAYAAAQzuocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEdvZWwgU2FtdWVsAAAFkAMAAgAAABQAABCkkAQAAgAAABQAABC4kpEAAgAAAAM2MAAAkpIAAgAAAAM2MAAA6hwABwAACAwAAAiYAAAAABzqAAAACAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAMjAyMzowMTowMiAwODozNzoyMwAyMDIzOjAxOjAyIDA4OjM3OjIzAAAARwBvAGUAbAAgAFMAYQBtAHUAZQBsAAAA/+ELHmh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8APD94cGFja2V0IGJlZ2luPSfvu78nIGlkPSdXNU0wTXBDZWhpSHpyZVN6TlRjemtjOWQnPz4NCjx4OnhtcG1ldGEgeG1sbnM6eD0iYWRvYmU6bnM6bWV0YS8iPjxyZGY6UkRGIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczpkYz0iaHR0cDovL3B1cmwub3JnL2RjL2VsZW1lbnRzLzEuMS8iLz48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOnhtcD0iaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wLyI+PHhtcDpDcmVhdGVEYXRlPjIwMjMtMDEtMDJUMDg6Mzc6MjMuNTk4PC94bXA6Q3JlYXRlRGF0ZT48L3JkZjpEZXNjcmlwdGlvbj48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOmRjPSJodHRwOi8vcHVybC5vcmcvZGMvZWxlbWVudHMvMS4xLyI+PGRjOmNyZWF0b3I+PHJkZjpTZXEgeG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbnMjIj48cmRmOmxpPkdvZWwgU2FtdWVsPC9yZGY6bGk+PC9yZGY6U2VxPg0KCQkJPC9kYzpjcmVhdG9yPjwvcmRmOkRlc2NyaXB0aW9uPjwvcmRmOlJERj48L3g6eG1wbWV0YT4NCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgPD94cGFja2V0IGVuZD0ndyc/Pv/bAEMABwUFBgUEBwYFBggHBwgKEQsKCQkKFQ8QDBEYFRoZGBUYFxseJyEbHSUdFxgiLiIlKCkrLCsaIC8zLyoyJyorKv/bAEMBBwgICgkKFAsLFCocGBwqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKv/AABEIAV4BXQMBIgACEQEDEQH/xAAfAAABBQEBAQEBAQAAAAAAAAAAAQIDBAUGBwgJCgv/xAC1EAACAQMDAgQDBQUEBAAAAX0BAgMABBEFEiExQQYTUWEHInEUMoGRoQgjQrHBFVLR8CQzYnKCCQoWFxgZGiUmJygpKjQ1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4eLj5OXm5+jp6vHy8/T19vf4+fr/xAAfAQADAQEBAQEBAQEBAAAAAAAAAQIDBAUGBwgJCgv/xAC1EQACAQIEBAMEBwUEBAABAncAAQIDEQQFITEGEkFRB2FxEyIygQgUQpGhscEJIzNS8BVictEKFiQ04SXxFxgZGiYnKCkqNTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqCg4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2dri4+Tl5ufo6ery8/T19vf4+fr/2gAMAwEAAhEDEQA/AOw+L/xp1H4beIrfTrHSbW9SW1jnLzSMpBZ5Vxx/1zH5155/w1jrv/Qtaf8A9/nqv+1R/wAj5Zf9g23/APRtzXiMsojkCrFHgKvVc9hVqKauxpHu3/DWOu/9C1p//f561NA/aU8Q6/fNbRaJpNttRpGkmllKqFUsSdoJ6A9q+cxMx6Qxn6JXefCwiPxnK6pGClsX+Yfd+XnH6j8auFNTdov+rGdarGhSlVlqoq56xrv7Q3ibQdbj02fRdGuHkVWWWGaUoQwBBGQD0NaHiz44eKvCFur3+laFcOW2tFBPNuT3O5RjPavKvGYW5+KOjG4EUqzLC7g9wyA4OMcentWx8YpRJ4ZtW+z2yvLLIzv0kJDKMgdwcnP4Vp9Vna7a0383ZbHCs0oynRgoP97qvLS+up3dh8dfE1/4eGrDTtBhRhIVhknn3nYpYjhcDIBxkjOKq+Fv2gvE/ivUJrOy0TR4JIlLZmmmw2CBgbQTn5hXM+E5ba08C2sEun7VuYFiaQ4zKJRzzjIGSe9cz8JJEttS1OTyfMZ3EH0Rjk9iOoHX0qnhZczimr9N9PUv+0aXJUqezdoO3TX01PRdS/aM8SaZ4kOjS6HpEsocJ5sc8pQk9xkZxzV/xP8AHfxP4Ys4bmbS9CvY5nZFa1nmIyu3PLKM/eHT3ryDxC0E/wAU7G4itljWaFZjFLkDdsJGcdOgrZ+JrCbwqjssKlZ3IRFAIbeAzcActgZ9wKj2DcVOLVlv56dCpY6mqtOk4O8/wPSk+OPiiTwzLrSaZoTRxRCWS3E0/mqpbaCflxye+exqDwv8fPE3ilbhoNL0KyW3VGd7u4mVfmJA5APdTWAX3fBlUWG1ZfsyAxpkdXP8R78Zx6g1znwsS2tvC9xdvZiaWTzNzvgriMZCnIIH3zVywkkparq+u2hh/a9F051FTfuy5baav79jtx+0X4lPiltCbQtIWdbj7OZfOl2bt23PTOM+1T+Iv2g/Efh1bN5NJ0S8jvFLRvbzTfLhipBDKCCCORXmVha203xgvXkt96qgughbIDna2eOoyx/SrfxXlS+bTJGi2tDci3HzfwbUfBAAwcuaTwzTi7qzv319DdY6m68aPI9Ve/by3PUdX+NXizRvDcWs3OlaC0U0aSJAs83mFXAIOCuDjIzzVfw18d/E3iaya4t9M0K2IkEYinmn3uT3AVTxyMntmuX+IUaJ4FuERYQkUMSqpxwoZ1UdM8BeOeuarfCBIV8Hed9lt/OWSZ0ncZYMqjH068+1CwdS3I2ubfy6HI86oPCyxUabsna3Xp5nTaF+0P4k17WpdOi0fRbVolZmmuLiUIABnqATTfEH7RPibw5rbaZeaDpMky7ctHLKB8yhh1APQ+lcN8OraOP4ma48cQBt1bYsTZVQeDjOcjFU/iBaW114y0aU2xja9n2Tqp++AwUHoMEj0p/VZKau1Z7b73sdUcxovEPD8jvZO/TVN237I9X8SfHDxR4ZsVubjTvD91mTymjtrmZmjbaWw2VAHANM0b46eJ9a0c6hDpmhQ/JI6wSzT73CKGbGFIPBHeuS+JVtbzeGbzzLRfOihSYTjAO7BXBwASME9fWn+DQtv8N7KGHyoxdWczvnncQjEk8HGcAHpx0qfqs+XluuZetjkWdUPqyxLpuzdrfK99zpPC3x88U+LLu6gstF0a3NrGZJHnnl27QQCeAf7wqleftIeIrPxJ/Y50XR5ZPMEYnjnlMZycZHGcfhXE/B/wAtbjV3a3ikLzLEQwz8pYEge2QD+ArF1fZJ8ZG3WsLhZJCIrc5Dld+PoeBkeoqvqz5uZtWdrHX/AGhTeJnhlB3grt99tD17xL8fPEvhm3hmm0rQ7xJZWh/0aeb5WUKedyjghgQR1q03xt8VDwm2vrpmgvbois8SzzGRCSAFPy4B5BxnpXk/xWmkk0LQ1EIih3SMoUKFB3Y4AA4IANddqao/wbRntIrdFtYwqkZZMk5z3wSBilHCzdo3V9L77GP9rUI06dVwdpvl6aO9u/f1Oh8L/HvxL4qWQ2+maDZmMoALq4lG8sWAC4ByflNUV/aP8RnxHJo76FpKSxymJpTLKVBBIJ4GccHtXJ/Cq2t4fBkl4bQTSyz7DvwQdm5l4IIGD+prP0PTrd/jjqKGMt5DmZAzdGyCT/48TiqWEk5JXWqXfqhyzSjF1U4P92tfOy6ano3iX9oDxN4Zls1uNH0W7S9jEkUltPKVKkkZyQPSrut/G3xVoWgx6nc6b4fkEkccgto7ibzQrruBIKgY/GvLvi64u30W7eMrJ5hiwSMbRhsYAHdjWp43XzPAlzus2HkxW5Vwq8BkPtkYCKOvTFZQw8pJO6899fQt5nRSptwf7x2Xlqlrr5+Z2/hz47eJvElktzDpug2gaYQolzPMCzEZHRSBnpk45OKq+G/2g/FHifWptNs9C0iGWJWYtLNLghfTaCf0rmPhc32TwB9qtrKJbny7hxcSEZYqBjB7DnH4VgfCSJD4w1qbZlo1YKcdAScjHI5wOtafU6knaLWu2/bqRLNaMFWlKD/db7a37a/5HoOuftGeJNB15tKuND0ieRcZkhnlK8qG7gHvV/xP8dvE3heASXOmaDdneUZbaec7SM5BJUDIweAa8l8ZQ2118StHxbhI7jYsidN2G2dunCgV0vxYWA+FZ1NqEmWSOcyrgAu+Q3QDJ471Kw7cVNNWW++uiZo8wpqrTpcjvNX6abb6+Z3Gj/HHxTrWhzalbaZoSCCN5Xt3mn8zYoyTgKRjtnPWqvhr9oDxL4mmu0g0nQ7NbVd8kl1cShcZA6gHuawPDMIj+G9nbw20hjuLCaVwoX53EeeMgk56fUVznwZjikvNY8+ySdcgZYZIy4GMenOfwqvqk22rrXbf8TCWc4eNKrV5H+7dntrrbTX/ACO4n/aO8SQeJv7EOh6O8vneSJlnkMZO7bnOM4qz4j/aC8R+HLe3mk0vQ7xJ2dR9nmmBRlIBDBlBByeleUXcMB+MpVrdEjjkkfysELldxHTtwOla3xbuo73Q7RvIEbW8yohB5CsgYggAegqZUOVKV1bXv2udUcbCVSFNQfvK9+34/wCZ6gvxr8VnwoNebS9BSBkZ1haeYSOA204+XB/Os7w5+0L4i8Rx3MkelaJZrbKGY3E03I9tqk+/0Ga59HhTwKunzWShDatGrbs5BRpc5AyDuGcZ71y3wl2pACLUSPLeBDNtBKj5cYyDg/MT+dP6rK7jdX+em3+ZlLNKSozrezdo6brXXfc9HX9ofxJ/wl0vh+XRtFhmimaF7h7iURAjOTnGccdcU/xX+0F4o8I3cNvfaHo87zRiQGGabABJHO4A9q8v8OzPJ8cZpp7USytcyMVODsbn5uOMf41N8b0jTWrTakaMYgSEHXOTn6ZzUvDyUee6st/U0+v0vrMcNyO8lzJ9Fv5nVf8ADWOu/wDQtaf/AN/no/4ax13/AKFrT/8Av89eF3D+VO6rBHtU4GUqL7Qf+eUX/fFZOMU7XO5pLqe8/wDDWOu/9C1p/wD3+evpDwvq0mv+ENH1iaJYpNQsYLp40OQhkjViB7DNfnrdKFunCgAeg7cV99fDj/klnhT/ALAtn/6ISoas7CejPnf9qj/kfLL/ALBtv/6Nua8Vitvtur29qG2Gd4492M7d2Bn9a91/aZ0y91P4gWiWFtLcMmmW5YRoWx+9ufQV5Vb+D/FUctvPFpJWTKvEzgK3HQ4J9u9aKMpQfKJyjFe87XPorwz4Yi8Bq2k6TawyK0ZZ5Lg8SY6yyN/COCAPfA7muG8EfZF+PHi1tQjjUpvKJ521VO8ZAORnjNZ08HxoubZryaXzZZ4RG0WIi7REn2xjPPXNXdB8F3OgeBJPFC3U9p4gkBhuFZ1dQRNt9D/Dj8a+SwWX1KU5e2qpzrWgnG/xPz6G9apTnScdLW67Ws99/wAg+IBsP+F0eEHiEfkZi8wJNu+XIPJzxwfyrU+N32dfhsixRxRv9sAI+0Fj17Ddyemc9Kq3Hhh9Y8a6fcahfzSzWmlLdQy5UMsnl78YC425qHU9Gu/Enh/w7b6zqMtxFf3jJOmVUKA6gbML15PWvcjgZqVH94/dXeWt4ylr8oPfyPPXsVyO8Pd8npstNNNWvvO2+H9rprfDfQppY0Nx/Z4wXnIzgkn5c47/AK1wHwCjtZZfFIvFUpvjIJlKAfM/oR7VrQWmpaR4Q1VNM1aaKPTbpLS1jyMKmcHPHJ96raZ4Vbwt4h1K30S/uIIn0z7W53gs8oj3DPy9NzHgUlgNK8VVd5Nu95aW5ZWXbSa2CXsnGV5x961vdfW++nk/mZ/jFLGL9orQ4ykaWaxRCRfM+ULhs/Nnjj3rZ+OJ0wfD8Jp6xhxej7s+84GQcjcec4z+FUl8LSarrvhXW7/UJ31G7kdJZww+VVbaNo246E9RVfX9Kvdb8K6nHqd/JOlrqSRQxkqFC4kOThc5+UVpDLZutSjGo/3do2u9fecNe/vRe/QcpUfdnKUdPJ9r6aab3O6e60gfCPyswqBpKMdt1jDMpHXd/eB47nI9q574GW2nzfDtmvlRm+2SLhpymRgdsgd8VLcaPem51bRDqs50+20pWjjITJO1cBjt6AsTVLwn4duNE/sKx0vUpra3vg8t0qsDubzWQbcrxwBXMsA54aovau0veveV0nFz/wDSV95MvYxThzRTbvs+6Wum99Cvo0NnN+0pq0ciKIBaAKqykAEImBkHmnfHSOxi0zQV0+NFDX5LMsu8v8q89TVJtBuLaO98Vw38o1k6h9l84sMeXsPbb97gVpeIvDb69e6xa6tqNxcx6PCbizDsAQ+E5YheRyeK7PqLVeNbndoLltd2urR29ZrcrmpKpF88dF2fn5ep03xKOnf8Kz1gRpB5n2Zdmbjnlj2zyeDgetYvwSnsk+GkaTiNpTczqR5+xtuMnjcOeDzxwKZc6dqGstoWk6lq081lq1puvEGwE8twpCcDgVmaJomo6D4It/sOoSWhfUTB5ZAI8tl5xlOvzGuX+zJvDuj7V3lLR3l3cbd7XgyIRoqk6bcO+zt0fbcX4bQ2E/xb8ZC62tbjcUYTlRjf/ezyMe9P+LSWS/EDwVHbIoj88GQJJnOZE754+tWLfwk+heKfEl1pmpyQS+WriQup3ZlUHeAPQk4GO1Lf+FDr/irw42q6jNcMtkboSq4BD8tx8uMZUVs8NFVY4t1HZR2u/wCTmvb0d/U1Xsvb8ylHts73vbe3fT/gHQ/FGDToPhnrLW8UXmeSq7hcFip3A9zzx/PNHw1bTl+FujC6WEytayHLT7eBvzkZGOP0rmNVtNT1jwrZQahq000d9efZrlOAuwLG3Hy5zljzVuHSNV0PTdds9K1G4jtdHKxWCYUjaXw24lMnjP51jLKZworDe1d+be8v5lC1992vzOeKpVKNlKD6/C7Wtfa3ZGX8ERaq3iPzUSQrfRlN020bctk9ee1Z2vLZyftKWyXAX7LkK4Sc9NjZy275T9CMVv8Ahbwm+ka9FBpmoTW0N7YJe3O0qWaby3bjKnC5FZMnhZr2bQ/EdzdyPqt7ePDNLuGAgIXgbcZwx610PBylipzVR+/FK13pdSs12+CR0RnQ9rKpzRt6O/TfTbVEvx6Gnf8ACP6M9isayG4kA2zl/kyf4dxHXv8AhXcXTaVF8JpI9qb10hSDHdEneUbGctkjOeD9Mdq4rX/DDa/pPiBtVvZZ/wCx7pUslLBQA0jAliFyTgfnWheaFcNdtojahK2njRvO2HbkusTMozt6Ak+/Nczy6bpQpe1d4Xbd5a2UZP10mtyV7DkgnKOj7O272030fzLPwPt7B/hmHu1Uyfa5QN05UN0wMZHvWX4djtJ/2kPEKzKvk/ZztUSkAHbHxkHmrvhTQJdD8M6DDY6vNajUbuRZUQqwA3AfLkdeec5rN/sCWz1LxD4gsr+VNUS7S1imLDDK24HcMdfkHP1rrjl7lXq2qO8vd3lo+Z09O3vLp0IlOl7zc42a/lfa+umumpL8c0s44vDSWgTy/trElZd24YXqc10fxLj0j/hWGqfY44/OEMR4uyxAydv8XPGeDnP4VkeJvCba5r01rq2oXFzFp2mC9t9zgETGMHnC/dz2qFdDvfEFroFhqV/NLa6uxS+CBRhUkKrtOzjAz+dcscDelSnGq7Q1er1Ti5699Ivf0LXskoJyjdeT7paab/qaPwa+wf8ACsraKSNWkma483NyUbAyfl+YFeAORj1rmvgolq/jPxYt4oWIgj55iuB5h755PvVi20S50HwfqK6VfPEtvqLW0UbFWXy2DZzlckkcVc0TweugeKvEY0m+uYBHYeepVgzSSbVYBvl6bmJwPatXgJRVeTqP3vOWluXbt8cdhydCSkuaPvW+y/PfTXYz/iTHaR/G3wpHbRqsIWEsqyZHMp75rqPjZFp6fDO/e0WPzGuYtxExY8tnGMmsO58Jy6t4q8IX2r3txLeTIGlcsFKEOSAo2/zp2r6Ze6/4Duv7W1aa5DalFAsYICbdu4nhc5zTjl38FKo/cst3q23FX76xe/QiUqKlG843X919ltppumdT4BTTD8NNFGoRIWexYE/aijFcfNgbhjj6etch8BBp6nxE1yEMwuVEYaXB259M887eauSaReaRJrWlWOpTLZaRZgWaNtJ52KQx28jDH9Kb4Z8I/wBh6hokGl389umsWxub0qyks4LEBcrwOBXNPAv2NSftXap7y1louX2mnb3e3oO1JqdO8Lt9n3trprr+JjEQt+07Zgr+58w4AnL5+Rv4ieOe2eOla/x5tdNtPAmnrpkaqDfAMwmMhOEbjJJNYyeHZW+yeJ/tjrrD6i0BmVgAF2g5A243ZY1t+LvDU2sjxHYanqdzdW2kNG9mkjqPmJAJYheeCa7JZfKFek5Td4rltd2buoX89ZLVj56MpRlGcbJdn2b008js7W20sfDmOYxL9p/sRT81wT0iODtz6k/yrz/4Brpv/CPXjXqq1x9uURBp9gxhe24ZOQOx/Wt/S9Lv3n0PSJNZuvsV5pe+aJSueEf5QdvC8AetczoXh6bw/wCF9Nn0i+kt5bzUGSbkMABtwRlevA/KsP7Nk4Toxqu82ne8tLuVlf8A7hy/AhSounNc0ev2Xbpvp5r+rlXwxFp//DSF8qiP7IskpT/SCRwvZt3PPYk1N+0JNZTQ6I9oMO4Z5Ns29cFVxjk4/TPvVtfB62Os+LNdtbuVL/TpgttJuBB3SlSWG3GcDtTde8Gprnirw/YarczXNvNDbeYQ6q43cHGFxgDpVrBv28a3tHaEXdXetoKTfZ6SXzN+ag6inzR7bO922tHb5HoFjNPP4A8O2cOm295a3FjFFdMoEgXKDCuOykdTXz18U/CFv4K8bSabZM3kSwJcpGx3GLdn5N38QBHB9K6jRtN+Ldtp4tvD9vcfYC2+MBoiCM8fePT26VheJNA8davrCzeJbKOS/YCJfNaJGI3EAYBH8Wee9eXlmV1sPjZuE04u7aV76u6bXS2x1161LlvJ2OHvP+Pt/wAP5V98/Dj/AJJZ4U/7Atn/AOiEr4Y1Hw5rVmJ7jUNPnhWI/vGaJgq9hzjFfc/w4/5JZ4U/7A1n/wCiEr6Kaak7mL11XU8g+OXiuPwv8QszWAvVudKtAoMjL5bLNdEMMfWuCk+LWm3dzbTXNjcJ5ZBkQQq+eSWwxx7dh3r0X40XuiWXj6c67DbTO+jWq2wuI2YBvOutxG0HBxjrx/KsMa78P/MtFDaXhUjPzwE4HXGRwDx0PQ4rfCRTc2o9N319NTyM2lTjCm50pT1fw3uvWwwfHzw6NLhtF0zVP3cYTJC54PrmtAapaa78MZNRSO4EdyGcKUGUH2o9eepP8q07jXvh8dHhZr/Rtz243JGAGB+g+YHpx1rOuntG+Ht01hMgtTnyWgOVZRdNjHXgf1r5nBqmsVQVOk4fvlvfXV7f1+B6lHlUZOMGnyLe/bbTW6621JYVt/8AhKIf3U7n+wk4CDp5I9+tUrRLb+zfCWIpwBfPtJQcnenXnirduwHiqHM7r/xIV6D/AKYj2qlaMv8AZfhD/SHP+nvwR/00TjpX1MU7w9I9/wDn1XJ5pcr93v8Az94/1/wLjrlbb/hHPEn7i4IOprucIODuPHX/ADmrtytv/wAJPf8A7uZD/YbAjYOnlD361TuEc+GfEarNNufU1KoE6/MenH+cVoTySf8ACVah/pNwVTQ2yQv3f3Q5+tQr3qf9v9/5KBpJy5V7vb+f+9/X59CpYLb+X4QxBcNiWXb8g+b5+nXiqVwtp/wjmtB1uNn9sR7tsYznEvT5qvWsqyN4Ob7ZM372X5nXn79F7dxp4V1bbq924/tREOIACn+s+UEnke/tXVSclif+3l3/AOf8/IfK5R+Hb/H/ACrz/r1uW7tbX/hJNbIt7kt/ZfzjYMY2jkc9/wCtR6fZkXvhRksLqRY4pM5jx5Y81zk4PH/16s3MUEviPXF+13AZ9OKH92NoGwUmnTR2Fx4Ytk1K6RnhcbRGMSr5knX0NebBy+qJLfkXf/nxM25JOalKOl3/AD9/u/rsZdxYXH/CKXkZ0K43/wBqbym1uV2t83X/AOtWtdx2i654jkk02aVWtR5oEoXzB8vQ54/KqkurRt4fuLiTV74suphRJHGqsDtPHPUVb1LUo21rxMGurpnhtB8pjjKqPl6cc/jW8nVc5JrTX+b+ej/wDdQjFppffzef9f0iKymszrfhgJo8i5tiYnM5/dL83UY5+vvWdaiCTwjb+VoE2wakflNwflbavzE46e1athJaTa94Zdri88x7M7EECBSMt97HQfQVnvdQx+ErbZdaiAdSCAMqq3AXg4HStIX5opJ7r+b+euS6tWM9acWteku3/D/cvMvTtZ/b/FCXWjzB/IjZvLcsZl3r7jGD/I1LYNpra1omNMufM/s0MjA8Km36/Ws+6k3av4uJvZUIhjAfZnaPMTn/AOt71o2BjGsaGUvyAdLC7TGu4jb9OnSuOsrYXr8P97/oHiYxqzdR+4kr9pd3/X/AM/TXsZfDOn+RpVyyf2s4TDplDhDuOc9scD0p95aL/wAVWZNNuAsjRnG9Rn5wfw/Gm2yQN4fsQutPEqauwVvKK+aQsYwQAO2fzqe+lgkXxYJNUuHClNylCAnzAccH+Vdlbm+se7/N/e/5/wACYSnKlrHpp8XZ/P8AruWNKtmTxNZubGUAaIgEmQRH+5b5Tz15Fc9aR+ZoHhkJauCNQc5JA/jX3rpbNdnjGyW3upZYzoqhhs4bET4PTg/41gxQSHQPDpV59y6k5KGMggFwc9OKxoSbqRb6xh3/AJa5zWa5rQdtf5+8fL+vvLUtjMuk+M/NsJ2WS8jKhesg81jkdasObBfFLeZZ3YcaI+4blxt8o8fXFWRqM9tY+LftF/dLBHeLskZR8gZ2GFyDx2qCbavim9kS+djHop+XbgOPKbnp2qOaTlU5u0tr/wDPugaOcYxSjF9N1Pu/69PKwtnYpNoHhZ7bS70xQ3MrFWA3R/OuS3Tg4OMVnn+zZdH16OWxvWP9qoGWHaWc7pMYz+NaOm3hXw14VP8AaUq755UJ25LnzB8ucHoM1TW68u012VNQmWOPU4yZCgYx8y8AEEeldNHm9vP/AK+Pv/0EyIlWUYXUHt2m/srpo/L/AIJtXtlBL4kvZY9O1Cd30lY8hR5bDyhhQf72MVSsbAQ6j4RUWN9A0UrnyygAjBlP3uTTr+6aPxjqRN3NGF0RXCp0z5Q5xjGai0ueBr7wVI2oXBmfdgiMFZMyHcp6Y5yK86kprB/9uef/AD4qHTKqpJRjCz/7ftv/AF5fK5jXKRt4X1nFnLM39rElccE/N6HPvW5JGi694okNhMB/ZJDOWwH+SPgf57VSi+zJ4a1eSTU54M6mxbyl5U5Ix07ir8k0f9veKkMtxhtLLMDjGPLj5HHWuqrU51Uils5d+9A537RKOnb+f+9/X9IpaciNqPg6RNNnC7DiTdkJz3qrGkZ8AylNMnCDWI22bufufe+laulGDzfB/wDptwMBtkJUYJ56nFRadfCHwKbiyvpbZm1RU3sFJ6Y2/d54rXnlzR5V9uHf/n7VDntK9RO3kpvovL+vW5X1JY11bxXnT5hmxjP3vvjdFx/n0q9ZYkv/AAjs0+V0GnE5Xny+W4JxTdU1e5N74ltpL+4a1hskKgIBtO6MZBx6H9auWb20Nx4VVZrgb9Pwj4XnqTn5c/liuKo5fU6d19nz/wCgY3i71Wo3+amvtfP+vI5aNbceE7FDpdwrnVjhGkxzsX5unTFa2uNB/avjXdZSOR5e5g/3vmX8qyhchvB9myXkpP8AbG1SyAE/IuRwK0tdYDVPG+Z2U/u/lA/2l9q9Osn9aX+J9/8An/SOZVJa2j07T/lf9f8ABsWNKkX/AITHw9GLEREaP1OSV+ST5etYccqf8Irop/s8DOpMOhwORz1rb0xgfHWg4nZ86RwSPvfJJz0rCjZf+ET0P/SHP/EzbjHuPaopL/aI+kO//UQJVKjg3y/+l/3f6/pmncyp9l8d/wChqdtwueD+8/fn3psksY8d+GwbMR5itcORyp3H3/zmkuSPsvj3963FymePu/vzUdwQfHfhkLK0h8u04I6/N/T+lZRXu1f8E+//AD4pFxnPTTqv5+7/AK/4FjN0f46aBpEEcFxpupSSRBUkZAvJUnJHNYV58WNEk1w3tpZ38cPmGTy2gjJBLlic8nnPrXb+E9d8GGyhh1O90dZAiJIt1FGOVHOSTzyTzVDVtf8Ah/8A2tI7/ZZCGJDRxRsvMjDIIP4/Qg15OXxpzzCfLTcZWV5XdmeXj/Zwoe9QlNNvRXfz+Z514o+JkWu6Bf6bHprxLcqqpI0pO0KytyMYydv4dq+tfhz/AMks8Kf9gaz/APRCV8s+NdU8LXml60dNWyFxME8gRwsHDB1LENjHI3ZGT0468fU3w5/5JZ4U/wCwNZ/+iErvlGMZO0bav5+Z60P4UNLaLTtpseS/GvQdM1v4gSf2r5jGHSbTyESVY9zNNdDkkHjgdB3rjh4E8I293DFMkbDcuV8xeV3YwSWGSTgZAxnpXTfHnwpeeKPiGq2VxHALXSrVnL7+Q01yB91WPbk9BXAP8OdVsJLS2uPEt2HlOwfZSzxpg44JZePTitsLKCc7u+m3Y8/MIuUYJVXDXp1PQT8LvA66FaXA0oO0kG4z+bgM3PYvjP044pkllDp/w4ubTTAi20I2oDyMfamxj9f0rDHwc1iSyiuE8bX5t5YlkViWxknG3G/tW0NKfSvhdJY3Ekt7JApRppB94/am5zk5z/SvnsNZ4nDtVfafvl8tdtX0O2DTjO8r+5s15b9Nyxbib/hKodvlf8gFeo/6YiqVqJ/7L8IbjFj7c/QdvMSrluinxVDm33f8SFP/AESKo2kajS/CGLbH+nvz/wBtE5r6aNrw9I/+mq5hpyPRdeke8f739fm+5Fz/AMIv4mwYdv8Aai9Qc/eNbSw3svim7CvAi/2QRuIONxiXGfb/AOvWM8CSeHvEa/Yy5Oqrkg8n5jWhJEF8UamWtSqjQjknoP3a1hJpe0/7e6L+Wh5nRFRbjzR006Jfzf3v6/JRFqMlz4WeeSCR45pC7E5x8/16U2a7vtP8N6gY1svm1RFUBWIIO85IPfp096u2dkv2Hw2Ro5co5cuHOI1LH25zkVn3ltCNE1Qf2SWRtVRgglbLcSfNkj/OauDi6+vSS6L/AJ/T8zptTd+WP4J9F+hr3NxfSa9q8LQ2LxixZldlJydq5B9un5VDbvqCy+FpFjsRE6tG3lDKqdzYx7dPxqPUDEmt69u0vAXTmPmb/vDaOMf561n20ludW8ICO0WOMxyYjDdP3j8Vx0Ic2DTS+wu3/QPPzFUny1OWKW/8q/m9f633Htc6nL4ZvGUW8Tx6ptJEO8AAHJPHT3q5qV1fLqniWJrmHyPse9FaMfL90Z9+9UU+yL4Yu/J0pJjJqeSjOCOVPPIwKuXtrEdc8SsbFSXtMFvOT5zlSOO3410y5eZtx3cui/no+ZcFOcVpfvaKt18x9jPdnXPCwNzbNGbTcUUgBiN2CFz6VC4vJfDtqjPZRlNR5MZZVAwD1zz1qKxlg/t3wpGbQJI1qR8qL8py3t/Kqapo48KWIhsLgxtqWNu/J83avfA49q1jD34adV0X89fzNKdKMpOLja3dLsvNnR3Vpqc8uuBDalWQLEVcjnzEJ3fN09KihS/HiLRonuLQH+z8YOS5+XBx69Kr3cQF14i8nRCxMMe5njJE37xOOnOOv4U+2Maa1own0lUzY8sIyPLIX7uccV59RWwuy+F9F/z4j5m6w0Yydl/5L53/AK+8rw6Zqen6Dp0UlzBNJ/abF2dW+6VTGM49D+dXb2z1i7j1+AXFnFuKLHIECnlgeTWLb39sfD+ksuiRsr6q6omCduAnz8j/ADir11e2KQeIxJopzGyZkVeZssMdVxwa768avtrpK/N2X/P+Hma0qajG6Wnbl/r+tDWttH1ePxPYXM14kkI0rymCE4aQRMM+3JzWNa6HrB0vQY7nyJngvWeQPITgbgQcjqeK09LlhOv6cDoxy+kjFzyu390Ts6YHHH41WsLXTZ9L0bGiSxAXZ8uMXO7awI5z36Zrgp1JwnDm/lh0X8tfzIdKlNuLSXlZf12LUelahJaeJoo7PTxJPco0fml9rjexy+e+PTvU9xbajbXV1cMumJGulsGK5Mm4IcDH90VFcWNta6V4iNzoZdJLhDIPNYGc+YTnjOMZB49aguLWBdSnkGk+W40mQCdhJgDy2+U9vb1rKcr+0fk+i/ko+ZpGhSULWV/8K/ruQR3N9Ho/hpFSzl8y8naRlhAUYcYK5xjvWXdXdzFo/iRIkgMh1GPKeR1yXPPPsOf8a11sYH0Pw4YtLdUiupcKcr5RLjPU55NQXNpYrpmvedp0RjN8jMokI3ff+YnNd+HqRlWnZfbfb/oIb7mChhFFKcYv1Ue3rbb5W8iS+t7i48WaiYnjBbRkjZWA4JiHqelJpmmNb3nhGN57J5IZHJAc5IMufl5xxn860tQj0x9euxJp6l/7MXzJEuCGKeWPl68DHGap2Vvosd94VCWk8TrM4twLndtPmchuOea5KdR/U0tfg7L/AJ8VPMuVHBtKVop/L/Pvb5mc6XMOg6sIDpLeZqJbElyNp5PUk8HjpWjNPI2teJIWGn7U0vg+d8w+RPvc8L7/AErOXTdPutH1iMafNcRnUyWEc5Dbuec44+lbL6ZZtr+vCOwkM9xpmCTc/wCtXagwFxx0xn2qq0qXNV01u+i/6ceYcuBcY3gnppt36a+pnabqFwNS8LQY08q4bJWUEjn+HnnrTLezvn8EsskdghGqJINsgKhQvXg9c9qng0eK31Xwuw06aLyflBM2Qhz345/Ssy30xo/Akkc2l8f2ukhjSXJxs+9ndXZCVOU4uGnvQ/8ATlXz8yZPBL3fZ/gtPx7WNe/02/vbrxAPNslWS1VUAcAhiYyS3PTj+VTw6Lc2+oeF5GubXNvYmNlCs28/NnaQMdD61U1C2j8/xEI9IZf9EVDLgMZMGLjn0/pUunoPtnhYSacJSNOb5uB5WGOeg9/1rglzfU4WenL2X/QN6nNJ4CM7Qgk772639f5tf6sYmm6RcX/hOFjcWlsserNJvcNsYbVHy4HqK0NXtbmfWPGq25hYN5fUHIOQcVlQW76l4atPsll5MQ1diEB4UbF7963NaFkupeLt1kxkwpeWUfK2SuOnbNepiJNYper07fvqRwxjT9muWC2etlr7r294jsIriPx7oiy+UWXSPnKdM7JOntWBGJ/+ES0PPk4/tNug9xXRWBjTxxoapZImdIwr8/L8j8D/AD3rm40QeE9D/wBGx/xM25/EUUf48L9of+7Byrl9m9F16R/u/wB7+vuvpXIl+y+PP9X/AMfK7eP+m561Hc+aPHfhnzSmPLs/uDn73H9Kfcqv2Xx7+563K5/2v35oa2aXx94aEFuVZYbVyc9ADkn/AD61nG3LV/wT/wDUekaRs2tFuui/mf8Ae/r8BPDvwz8Gahp1rc3+kRs0kcbSM0rDczD2fuenH4VS1L4eeDbbUXhTR0RFdhkyjAG4jvKDwBnp2rJ0z4M3+p25mHiy7gEu12hWJm25JxwG7GqN58HLsXhT/hKJLlvMKlihByGIP3m+p/E1wZar4+a9o5aL3ei9L6HnZk4RoJSxDh7z1s/u0fQzvFvgrw7p/hjVNSsQYbiLy2t4xcK6sGZQQFxuGMkdT06mvqj4c/8AJLPCn/YGs/8A0QlfH/iP4caloeiXWpXGpwTxwqGaNHcvgsqjOUA/iHf6ZxX2B8Of+SWeFP8AsDWf/ohK2dru0r6v5eR7cf4cNb6LXv5nif7QWra1pXxCj/sGaaIz6TbpMIlzuXzbng8Hjk1xlhr/AI5eGzmfQo7xwFMUs0jKXweCVBA/SvQ/jf4rg8MePpVuY7llvdItED20/lOhWa5Oc7Tnr0riB8V/DN3NBNf2l2rxkZCqHxye+Bnj2HU114Vv31ZL1a1/A83MVJxhal7RX+a/FHQR+IfirFpsNvD4O08RJCBGVlBO3sfv+1XftmpSfDFpNYL2d75ZMkcT8K32og9Ce2PzqpF8dvCiaZDbtHqBkjhWNmEAHIPrnpV4a1BrHw1l1G0VfJuFMgEw2k/6Uw7Htz+dfOYeNSOJw/PTUP3y267/AJnoUnKKm7cvurX5ej29GSQzsfFESteThf7CRsBj18kc/WqVpcudN8JEXtwS984bLHn504NaNvP/AMVRCAsKn+w1OST/AM8Rx16VStZ86Z4SysGGvnAALfL+8T3r6SO8NOkf/TVfyI9rLlfvd+vnH+5/X4pPstlc6D4lW41LyC+og7nDFVO44Bwp/StWe1Da1eINSJibR2XyiH4HlD5vu/WqIuAuh+IC/kBF1VPlyeTvPJya2PMefxffN9ts4X/sok73cYzGCG9MDPbmsJSlzVNf5v8A0ij5HXCpKSUU9Wl1a7/3P6/PMsIYYrfwug1OVPnfK4P7wbzx0rPu2lHh3XPtF1cjZqse1hGW4xJwM449637e3uoJPDUceo2csETylnSUFWJJxjJye/rVe5t72XwtrMb3cLtLeqBlgSFG7A+91/GtKddLFO/88f8A0/Py7Ft4iMbJydv73kv7nf8AHXyHXa2R1bWWF/Oso00ll2ZC/J245/Os/S1LXnhENPKIzBKXYIT/AMtJOvHFbEt7dWviLVwbuIpHp29YpEynCgZ68+9XtO1q4kj0UfbLFPtMTbkRAN/zuBt56ce/euJTmsJp/Iv/AExPyNov2lT3qjVul+z9Pkc4sNnL4ZnzqtxCraj87Mh4G08YC9DV3ULS1k1jxL5l1s32oBAydp+Xn7n9TVmS4lGjymPUbMk3gUuUG1BgnBwDzU19fkaxrsY1C2DQW4YJsI2/d5Y7cfz61v7STTf+L/0uj/dO7kipW9pL9Py/qyM6ysbZtZ8MML5mMducKUI3n5ufu/4VmW+k2B8JW0J1LZjUy6s0bEHhcjhc5rooLky6t4fEl9ZEtGXCq2C5+blfl6VRttV8vw3BJHNp2035jJHQ8L+tWqlVONr/ABR/9LreQOmuf4n9/wDwB93pulvP4jb7c4eSCPzeHOwb0I42eoHTNFpaWtrq+i+XqEpkFgIz5auquAvB+7+hrSkutUSXXdkmmLGEQwlm5B3KMvzwMZ/HFTJfatJfaYsk2nPHJAGlG8Ft+3qPavNq1Z+w5b/Z7/8ATmP90bjFvd/18jkto/4RvSDa6hcHbqp+/v3PlY8j7vQe/FXrvyXh8TK2qXCkFcMQ+E+bjHy/TpVyPWL/APsfTXM2ieZJqDRvtK7MYTp/tf8A1qS61ieH/hIGaXTSLcIV6n+L+KvUrOpKrZfzd/8Ap/DyMnFQu+Z/18iWxmjHibTI4tUuHL6PkRbm2sPKOH6Yz39aw7V7ldG0Bv7TvS/9oN5nEmXG4cHjp9a6Gy1CZ/E2jxrPaBG0rzMRYyD5Tc+uOlUbDUNRtNP0TdcIZJbx1fzHBLDcvdTj6Vx0XKEoRWr5Yb+lfyOD2lpTlK6V337r+6MKxyWPi77RfXqxm7T5lLFo/wB42Nvy9O3Gaey248QS/wCl37f8Sd8BS2P9Wcnp970+lbD6lqMdh4guInslmimUREkYxvI+bn0p66jfHVNgvtOVWsDLs3Dfu2H5sZ+7n9BXPXqyUanpLr2p0fJ9jrppJ2Umvl691/X54yw2D6D4bxfaiyi6kMbNuy53jO/jt2rP1K10k6P4gDXOoMh1GMyNjkH95jbkdOT+ldbHdagdM0k/2hYBnkk81t3DgEcLz1A+tYut3F8ui6wZ9Y01j9sRVbyCfLX5/lYgEk9MHtg1vldSTn8V/f7/APUQ/wC6v6+4dajdtuTVr9F6fy/0/uG6ommx67qE/wBtnS7GjqPKV8EKIRg9OD+NUtIurdrzweRfXcheWQJmTPmfvjndxyK0dRhd/EV6wvLYD+yFwCr5B8ocnjp39an0NZ7L/hG1jurKYOzq52jLKZCcrnkemKUGlgLXbfL/AO4ankEqUHaSm18tPyMFYrYaBrYmu761VtRY/ICWY88DC9K0p1tzrHiJpLm68xdJZDGy/KibIzla0LvUdSn0m/Zr3T42W5aNMjcAvPBHrUF9K0mseIbZyoA0klWQgfwrnqcDk1Mqk3Orpu31/wCvHl+pyYijBKHLLb/g/wB19/L/ADqab/Z63vhSKPUbzbsYxo2drkEcNx61nWi6angJy95Oyf2tHlg5BB2dPudMe341o6RYSzTeF7lr3iJdvlmRCWyc9jzz3FRw6C8Pg5oRqEbkaqku8SxsBhcbeuPw613RlBTScvtw/wDTlXyGsJWspJrbv/8AaeX9dF1K7s4dQ8Rot5fyhbFGmUS7QuWjGVyvBxipbOWzbUvC7C4u4mfTT5Y3Z3jcfvHH9KmuoLmW/wBejkvopVltIyEcRlUP7vqev58VetdNENx4flW+tQY7Py9h6vkk/L7c/pXBKcI4SCvry/8Auv6foOeHrOreNmvl3v8Ay9vx89TibeaFvB1kEu7mFf7WIABLk/IvHbitPxAynVPGoa5m4EeF5wPmXpUNzazaf4VsyLqxZhq2Q1ux7qoxz39fate/1S5g1zxaFkg2wvHtEi5xllHIr168v9pUo/zP/wBP0vI89TavCpK1l/7a/wC5/X4NliIf+Ey0ArdTyEaR8ofPI2ScnjrXMR+X/wAInof+kSk/2m3HPqPauujnmfxxoQe4gk3aYxYxj7x8tzn6c1y4MieFdEVpYjt1RwcH3FYYOo51ISf9z/3Y8jKf7vmjF3X/AOx/c/r8rlzs+y+Pf3r/APHymf8AZ/fnpUcrAePfDPlTSOTFaDDE8jdz/n2qe5Zvsvjz504uVx7fvz1q1a3BXx94eQm3LNaQDeck9DwOcZqo35av+Cf/AKj0gjUel31X/pT/ALn9fi8HS/EPxNtUEeh+GrW8sY9qwvI4yydFJ+ese91bx9Pqxu5/Cdil15hGVlH3txyPveua2tF+N/hbSLaO3uY7/fCqo3lQgrlc5xlqzL34v+GpNVNxbtqPl+YXAMWOrljn5uetefg7rHSc6KSt8Xu6/ecONhW9i/ZUlJ3en66dzzvX9e8VXdtqEGrNMsMuFuIygVRtI2j7o6YGOa+zPhz/AMks8Kf9gaz/APRCV8l+J/iNpmtaDqVnBZ3S3F4iIsstxuVFVlbAXaOPlzyeMmvrT4c/8ks8Kf8AYGs//RCV11JXk9Lavtr56Hqxuqcbq2m3byPKPjOmiSfEQjXpbSL/AIlVn5JumYAjzrrdjaDzjHWuVudS+HkF7bxQx6VPHGEzJh8EAgnp374P0NdZ8ZPCVl4r+Ikq39w8ItNHtXQJjLkzXQxz34GPrXCj4VeHI7qNZFvZAWGRG8QGBxg7mBBJx0HBrXBzSlNRd9NU9l6aHlZs6EKdOVecoq7tyvf1Own1f4dnR4GL6CSLcDYhG4Mck9Pm9OvIqvKLR/hxcf2ekU1qFxF5TZQL9qboe+cfpUc3wW8HrpFrcJa3jvJbgkrMvzt68tj04HFPlsLXSPhvcWFqHS3gBSMKdxwLo9T+P86+ZwU6MsVQ9jJv98r39Xc9Ok04yUJNvkXfto9LO5Pbxg+Kof8ARg//ABIU7j/niKrWEEbaR4TzbBGF65BMigZ8xPzqeB4x4qhy0o/4kKdAf+eIqjaPGdL8IYabP29+oP8Az0Svq0m+T0j3/wCfVYUVVUXo+v8AP3j/AHv69L30IoLefRfESnTlw2pqCzzIMnce5xj/AOvV690x21y+lj051B0h4w+V2nMQAXOaybue3TwzrogWaSQaiN6lWxu8xun4VpPqbw+Ir2MNcKi6M0gALDkRD9a50qjdSS/v73/ko+Z1qnKUFz36fzef97+vusml2iR6b4VD6UrsZmKuXXK5LHH9fwqHVreC38KarJNooRH1GMkF1yxG8buOn4+tTWepJNF4abbfSGRnJOScDcRk/wCelMu9QgOh6i9xBqNwqalGrqCwYnEnK5HT/wCtSpxrLGNtfbXV/wDP+fmNYeq1FKXT+9/Kl/N5X/4N2X5dPeXXNY8rQ2czafhJcjD/AC/dHbJqvbWF1Fd+FVk0yJTFGVlGY/kO+TI/Hg8VcutTii1bUwp1L91ZMxXewUfL2GMA+4NV7G50wyeG2mm1FZpI3MKc4zvf73HJ61z05VPqm32V3/58T8zephn7Re87+skt79/69NCpJpd1e+GHgtNItxP/AGjlwWiO8Ybn0zVzVLDydU8Ryy2KsslkRuZ4wCflGeOfzqk0sMfhm78t9QVmvz1SQ87TjgAEin6jKqa34oknW8KrZgBskKfuHC8cfrXUlO8l0vLvf46PmTOnJRslJt/4/Pz/AK08iKytVGs+ET5Mastp18wcj5+QKgt9Pgt/B1ulxYqQdT3AmZTtOF6EY646e1aOkNor3Xhlp5bz7WLbECqpK4+bO41WDeH5/DlqH1HU2gGoFlcxEsH2r8uMdPejnk5xhZ/Euj6VKwU8Jib86XTvLqrfzeT/AOHuXJrUHU/Ewj0q3ZnijG1nU+Yd6cHPSr1hYLHc6UZdIjZ0tFQzb4vkO08Y6/lxxWdd3enQXPiaQXNy7LEjSL867fnTocfTpV601GybUNLCTX2ZLRWQfPsxtPJOOv8A9evJxCqulonbkX83/PmPma0qdWk37Tvpbm67df69CPTrSIaFaK+h20wTUGIxNCFiJCfNkcE/rx9KJLQedr7HRbNgxQbTcx/vMN/ED90/WoLTUNJl0PT2ikvXifUWU7fMJLhUPPy9OnWp7vUNPC62PPv0MezzGIkOMsPujH8q9Ospuu1Z/F5/8/4eZtGM7O79Pi7ev9epatYgPE+kEaTDGraWQZEZCSPLPyDHOOcZ6c1QitVt7bQwvhqKGL7Wy+TLIrGMswwwJ796tWtzanxXpPl3V83/ABKGYpIrbSvl5DZI64zn3xWNNe2k2leH5LeS/XOofKZAxP3ucmuGlKp7eEUnblh37VvM437WEZyu27u3xf5/5eh0EyWhtPEO3QLYgTpvjaaPE58xvmb09eaZNZxDXTKNEiDJpkiOfOTkFGwmP0zWYL2Eaf4waW7u9qXqA43Box5rDA4P6VOLyyPiZ/398s39jM+QWww2Nz06+/rVSpyiqqV/hl3/AOfdHzG8TVqKLcXuu/n5+RPHbxto/h9k0WGNVuJcRtJH+6y4GRxyT7VlXHlT2uvIujQTD+0YwYFdVB/1nzE44Pfn1q6b62fSPDRF5fmOS5lwWL7nIcfe459s1gXhtRpfiR5LmQqdTjLEBwVP7zj7prpyyL5mnfSfn/0EPuycVKu37t9b/wA3b/F3/q+p1N9HaHXr0vplu7/2Uu5zdICV8sfKQeQO27pRpv2AzeGQNJt4pNzNFsuIj5X7zqOMtnr8tUNT+x/8JPqWZ5lc6IoKgHAXyhz09KqaY9mL/wAFGGe4fMjqmQef3vOcjpmsadNvB9fg8/8AnxU8wdavCyUfz7279jUuEsBpd/J/YEE0Qvf3iNPFtY8/MT0H4881emlsv7b11BpNorpp581/tEe6UbV+VhjIGMDJ44rm7m70dvDeref/AGgif2htm8oclgTjHtwKuXK2v9veKpY5bgynSTuV4zjGxOQe56frTlRlapzX69/+nHma1sXWio2Wj9f66IWBbUa94VcabHHuDeWyTKypz2wMH8PSqmnn7T4ObydJgmSPWFzGiKARs+8e3HrTdKa2a+8Gh5ZgwX5VYNgnJ9jViK80+bwQALme1hXVI1x5byHO3O0DjGfXpXarqUVZ/FHv/wA/KpharNtvTy95dF/esSasEttR8Ru2mIB9hhYqQhDjdGMEjnt39Kms5oZNQ8JJ/ZaM8liSrAgCPlvlHFVr77HNqHij/j+djaxhgEPIHldDzk//AF6ktjFHqvhBWiu022BKnJ+XluGGOv5Vwb4SC68vn/0DepnbEKpdXtf+8/tettv6sZtvLav4ZtVk0RAG1R0VRIo2sEX5s4x/+qtDWII5tZ8YKdO3nEe5hKBvG4c9OP8A61Z4vNIk8I2Riju9jamRtbnEm1Tk+orU1W6tItX8Wg/aVZdpkMYPJ3AcfhXoVnL61on8T7/8/qXmYSdblu7t2eyl2l5/19zJNNjgi8TeH4xYBZP7LISUSKdo8tvl4HP1965gRbfC+i7rNVb+1Hzgj1H8q3dMkifxh4c8t5j/AMSfjcDjHlN+vX9K56J4v+ET0MBpv+Qm3UH1FVh48teNv7nf/qIMl7aUG2nfX+f+7/e/r7zTuUH2Xx5+4HNyncfN+/NPiiH/AAsLw3/oY4toD1HHDc/1qO5ZPsvj7mTi5TPB/wCe56U6KSL/AIWH4a+ab/j2t8cHrhv0z+lKF7Vf8E//AFHpBKNbl2e6/n7v+9/XpZEHhbVPAkVnCdRutGFwFjWRZ4lVgR15JIPJPpn0qnqHiD4enVXZBZsA5AKWcRXmRgDnPTgfgQam8N/CDwrqltb3d9p905nSN2BmwCWGSRhs8npx2rK1P4RaBFqzwW9pfxwiRlD7o2UDeQDkvnAA9M14+WSpf2jP2cpOVldPb5Hm5lWwqw6dWpKKu9nrvqtjF8cN4Kfw7qs2jS2D3kwVoViZg8Z3rkAdORuJ9MV9QfDn/klnhT/sDWf/AKISvlPxb8PtF0jw7qOoWMlyslvFHLGkgG0h2UYBxzjJz34r6s+HP/JLPCn/AGBrP/0QldrabbTb1e/9bHsxsqUEtrL1+fmeKftB6Bea349V7O4ht0tdKtWlkmfaAGmuQOx9PwriF8FeLLP7HbHxTdRGQBUWBZZEj5xjcvAH5Cu4/aA1PWdP+IUY0KNpTNpVsJlEQcECW5xkEH1NcJB4h8eXLWty2gido9vlsylQ2GzyoIHXg9K68PzNSV7dt/xOLG+15Yez5d9eb9DrV+F/i1tHimTx5ehpEDh2kcRhSem3OQa010650z4Tyw3moC5uo2aKWV+TIRdE7uT7kVj/APCZ/FdLWOH/AIQ+22JGMARHlexxuxW1HJqFx8Jzd6vp4jnk3edDv27H+0np16gg/hXz9F13icN7Zxf75Wt89/P9TpipOM+ZpLlXbs776W/Afbsw8VQ4mRf+JCvUf9MR71StWf8AsvwhmeM/6c/AH/TRPetCAqfFEO2zy39hrz5vGPJHHT9aqWSK+m+E8Wu0C9cqTOo/5aJ7c/SvoU9Yeke3/PquZKnFxdpR69Yd4/193WxLIJU0HXtl7DGG1RCCByvzHrz/AJxVu7WWTxVqIluleNdGYBwBtBMQyetRQ3LJpmuoNIiuAuoKoSSQfPuduc49as3lzcQeJr+N7G1VF0d2KZBORGOD7dq5lzqdTb7fb+Sj8zb2dLlSco9P5O8v6+Tt1H2EksVl4bQa1DG25l2IP9cN+CMg9KnlENxpWo/aNctEX7eo82QNhTh/lOJOv5dOlYtnLPJ/wiTC0hHmTS8DHH7ztxxVe5kkPhjWmNlEyLrC8Lt5GH5PH0/OtqdF/WPi151/L/z/AJnZSxUKKXK42t3jbRLz/L8jrpYIRrF/J/bdojfZSGjOcxjH3z8/T8qVXZ20V01WwmGHB/dHMnzMMp8xx6d+hrFuZCPEWuBrBX26YWbDL8y7Rx936VDYXIfUPCaw2SIJIZMKzKdp8x++3/DrXFTov6qnf7C7f8+Js7JY2N0uaP3r0/rzH3t9ezeHriU6ppx/00KGTzdoADcHjOc/hWjqkl0LvXRFqNswNvkREuCh+Xr/APWqjPHdtoDh9DglaTUPkjTaoPyt8x+Xg/Wnahb3Q1TxPMmmEo9p8shUEP8Ad4+7z9K7Gou+28u389E1njI8zjeO3df1bT8yzpU7vq+gxnUkRGtMiMoR5hyw9f8AHpTJCttocG7XIZAb3HmRx5BOF+XG/wDX36U3TTdR6x4YiWzChrMlgHXCjntimQyane+H4jbafHBImoMrKXTGABz93/OK54tSqxkmrXX8v89YUcVeXIpq/bTtfv5l65kTztfV9TicJGgMbxAqnzpyfn5H5Vbt2zfaf/xNLPH2ZfkEfL/L1HzdOPeqtzPeWs2vGbTraVBEjJ/pCDdl1G1uPl7nn0q3B9se+sH/ALFt1U26lnFypMZ2/dAxyPevOxEX7BSuvh7x/wCfMSo1eaTi5K/y/wA/69SnbNbvodiY9W0/yvtx5+ylVfhPlC7+v+PSpbiSMDWd2sWildu4iA5i+YcHD5b9KitJ7ybRbRxoVuGW/ZZEN2h8oYT5sgdfbrx70+8kvEi1ll0iBtu3aqXKEyfMOox8v416E/8AeWr/AGu8f+f8Be1jyNqUdPT/AD/pfeOiJbxDpzfbYJVbSn2qFIY/uj82CxwD/TrWdb7l03R999aSqt4y/cyCd4wAN3+Nalld3L69bRf2ZDGE0gSNKJAWQmM/L05HaqVtHfDS9LMuh2oJuzvBmUeXlxhgSOfpXnQkvrFOLsnyx6x7Vf8APp+qPLk37751v/cstV5/n316El9LO2m+JBHf2rSC4j2iOBV8v94fvFsBj9TUot5zrBddciVv7MZdnlICp2H5t3oOuOnFMuftAsPEbHRYGzcoEjE0Z8/ErcnA4x70kupiPxA9k+h24mXS3cTbwTgIfk6dOMV1SjK1TlttL+X/AJ90TqpSjLeUf/JfNf1/w4g0+WTS/DzyeIInkt5pf3uF/fAsPlHPbFUb+e8i0/Xt2pQgC/j2SJbq4CnfkFQfw59KnGrr/ZHhndo8QN1PKFQMD5WGXkfL3/CqV5dmDRdcafS7Rdl+nD7QHBMmGPy9f8a6cBCrF2na/P8A3f8AoIfb/huxckpy5lb/AMk627/18jV1Azf8JBfBLqJFbSVADQLyfLHXnOO+MVX06xt0ufCYOpW0jxuzLsgYb/3uSBzx6c1avtTjg8Q3Q/sq3mZdKWUvuGWHlj5fu9O3XtVewuZLm78KTQaNZ2sUkrEgsn7keYfuZwST14rloqawe1vd8v8AnxU+ZFSnSaV+V/8AgN9/l/w9uplTwtP4d1i3W7sYPM1Xblj0ye5zwf8A69dBLpdxJq2u7tUsSkmnGNE8z5osog3N6Lx+orH1GaO38O6rJ/YVq0Y1D50Nyrh+vzHHQ+1XJLuCbXvEkP8AZVqhTSyWm8xg0o2J8reg+npXRV53Go13l2/6ceZyujhU4+0cW+msfPy9fx8yTT9KnjuvDS/2ppzrarhlBBMnP8B79axXjlg8Hy/a7uMN/bUZWWIKy/6vjoav27JFqXhHbo8MSPzG6zswj+nr+NULZmPgucxafbxFtZT90zlhymc59a3oX5ou905R7f8AP2r2M8RDDuajHluk+sOy7/L+rG79pmTWvEK/2pbIptkbc8efLI2cn5unJ/OlsRdm48NMl7btCbEiRnddznc3IG7kfTNQXEVxbap4huG0q2cNaLhWmz5nKDBHYDn8qIDNcXXhWVdOt1H2PLMGx5f3/lA9O9eSoqOGTVrOK/l/6BhWjGolNq99rwX2nb+t/mZiwy6X4btyNUtSyanyyRB1b5EG0c4zx1q9qd1M+teMY/tFvCIlQLmMZHzDqc81z0MMh8JQKbeKPGr5O6T7vyJ3ra1yOe21bxjMbeORJBGwO8cjeOD6GvZqxX1qzd3zPt/z+onM0p0fcUUrd4dpf1/wLk+lq0fifw5/pMTj+yiNoUZP7pjnr0/wrnhLI/hXRGaeLnU24VQO49K3dKnaXxV4cX7KqAaTuLhun7pxt/T9awYHlj8JaJ/okJB1JuWwe4q6Kf1iN/7nb/qIOenBezak11/59/3f6+7yL9yW+y+PP3qf8fK44+7+/PWpoPMf4ieHMXERxaQE8AZGG469e1NuZpPs3jr/AEaE7bhcfKPn/fHrUtvJMfiB4eX7BAwNrASeMoMHkH9azje1X/BPt/z4pGvs6bW63X8n8z/r/g3OR0f4UeKtQt0mtPGbWizlW8pd52ckrnBxxzg1h6h8Ntc/tPyX8RTX0rSmMyrAxAO9lOSzDvuJ+p610Ol+Lfilb2scOkeFYngQKsTLESSoPH8VZOoa38QZdb+23XhWOK63YxhgM7icY3eufyriwaxH1ySlyuFlZK/MvWyIxU68aWk1F3e+3lv+JyOveCdY0rT766udQguIrYAy+W+SQWUDIxwfmU49CK+yPhz/AMks8Kf9gaz/APRCV8deIPEvi690q/g1S2aKzuNvngQBV+UjbzjPGAOtfYvw5/5JZ4U/7A1n/wCiErpqX5nfvpvt8+p3q/JHm3sr+vkeSfG3xbH4V+Im+W2km+06VaBTHcmEoyzXRByFORz04rin+I/hnVZra61KQW8iD95GYDKRySRu29P/AK9d38ZbnQLX4gzv4itYbljo1qLZJt2A3nXWSMdwMcHr7da5mU+Avtdp5Z0/YVjZszSptAIfgBSPwJ9q0wkIt1Lw6bvVP09Dx81cFGk5Rk3d25d16+Rqw/GbwXFpFvave3EkkMCp5n2Y5JHUdOlSyahZaz8MpL62uJfIud0oyrDbm5Ocgd8jH4VeuLvwNJodsrz6IZEt/mVZRuBYeoGT1HUZGfwrNnWCL4dXKWU0SQKNqNG29WAuj0PfHr718zglTWKoKnCUf3y39emh6lFQUZtJpuC3v2fa7v3sieCOD/hKIiZnJ/sJBhQ3TyRzVWxjh/svwoEDTYvnwxkKY+dOxIzU1uW/4SqHF0qf8SFOw/54iqVoW/svwhm6Vv8AT34wP+eicV9Uk7w16R/9NVhKVovXv/N3j/d/r72r6m3fStczBc7RqKbtszDJ3tnHPFX54bM+LL2WVZDG2kOGDSneV8sZ7/rWWTP/AGP4gQanBBt1FSjMVIX5m4OM/rWw9q8ut3U82sWiwnR2UghS6/ux823rjvWEk4ynd/zd/wCSj5G0abqK13ey795a/D/X51tMGmi38OCW1myHk8hkmJ2HzPrzWRdS20PhrWWNrM8Z1lA0TyMuflfOCPcH9K1LcNAfDKJq8DDzJAGWFQH/AHg5Hp/9as7U3efw3rCyajG2NZAB2KAvEnp61tQ1xmr+2u//AD+n5CrSdOCV9k+/RLtH+vwNe9uLYa3rISwmMq6YSX89/mXb93HYfSqelyWZufCYaznWUxSlAkxAQeY/Bz1q3PceX4h1r/iYRKU0zJHkgkDaO/fHp7VXspkfUPCciagAzRSgEwqN4Mr8+2Mfp71zU/8AdP8Atxd/+geZpUnqm2t/7383lH+vTUaiuvh+fzrW/P8AxMclY7tgxG1uQc9Pard4d2u+JI2gum/0UYVbrr93oM/LWW95cHwxNImoQCQaltViiALiNv8A9Wa6C+cC+15Z9eiRTbjMIjVjCMLknBz+frWs+ZN37y7/AM9HyNaU4VJ8qb2/vefeP9fdeCzjT+3vDbra3SqLVgC1wSU+9wfWs8xQr4UtUvoNRWRtQJRVlZjuwvXPartlIz6x4bK6tZsn2b5EkhIeQfNgr8vy/mOlZQleTw7bM3iGxDf2oX81EcKx2r8uNvX9OazwtN80W31X8389fyO6WFlJtxSb6e8+1u3l/VyzehDJ4wRbe5MjiP8AjbDDenT3q/aP5eq6WI7e7EZ08YbzmwMA8MO/b9apzH/T/FEja/AmYY3IVHYwHenJGAD6cHvWlZXu7UNLjPiJW32KkxeSAJQQfmGen0+lc+Lc/qqUV01+L/nxDyOWlhKlGrKUnv5t7u/Y53TZQ/hOxlH2rH9qtnEjcttixuPcZ9aszOkVz4sJjuGHmRHAlYHBfpxU1nIsHh6xhsvEFusb6mwaR4lQS8R5Qbcj8felvFUP4llGtQFnaItgDMZD/lXoz1xMr9Zef/P+Hl6mUcJXUXy7+suzXbv/AFc0tN8tvEtmEhuFkbRV+cyMRjyjgEZ5PWs2Ap/YujQzLfu7X5CFpW4w4+8O9aOlzSHxRp5j1QMDpC5jwvzfumw39fSqSX12lloxfWlZ3vGy4RD5gLDj2615VOPJUiu0YdXfat5BPCVpuXO0r+b8rdPJ/wBNjrkWs+leKkEF04F2gkQyE7m85vu+g+lHmhvFW3yJmYaGzZ85gfuNwOfWrz7rex8SSy69DEftCuHUI3kHzGwCM9T05xTW1BjrvlrqyszaS7f6mIFfkJyWzn3x0rW7l7RLtLv/AM+6PkZ08K6Wjk7K383eT7ef9Kxnu8EeneDxJbXOftEqqpnO5GyvOe49BVXUbi0bw74gFxb3aoL6MbTcM2W3SYI6YHtWq7XMml+GGfUY5j9qlLyjyefmGMc9v9ms65eK40XWxNrEcf8AxME5a3U+VzIMHjBzj9PeunB/F73Sb2b/AOgmRdSLVrNWffmXT/D/AF66GpqYgGt3ziG5LjRVyVlIGPLHGM9f61lab5T3vgxvLl8w3DlC0hOF81uuevauonDjVLhk1y3gRtPT915ab1/dj5znBx3x0qvZW3mzeG5G8RW8rRyEhGRAbj5+gw3BHTjNcVGqlg7P+Xz/AOfFTyNJYHnkpRfnu+9+3l/SOP1GNIfCutA+Y5bVM/I5wOvBrUkVF8SeKiGYt/ZBDLz8v7uP8KsfZZ103Uydb021LX5YOyRMq9RggEjP15rTu4H/ALT8QSzavp7JJpxBjWNd8Y2p8z45I6nHuK6a1dKNT1ffvh/I0eBk0mn07y8/8/60MWxVHufBZEc/yjkvNx0HY9aoIGXwC/kW97C51lMqXJY/J16A4rbsg327wso1qwmRMqqCMZk6fc9PSq9lPcnwkzrrdhHL/ayL54jCoBsxt4GM4rWhJrk/xR7/APPyr5Gc8O1KzevrLy8v6+8kvZ0kvvEkapeGRLIDAmbp+7xgdjn096dZ8XXhMyxXRYaf87FjheW689c+vapr+GX7ZrxOu6fmWyUbdi7ouU5c+hyf++qltrWWK48NldVslK2JRkVRmUZblfbmvMlJfVI69PP/AKBvQr2Eufmcr/OXe/a3l/wNTmbK3Fz4OtPs9tJO39qFmAkJx8q9cN6etb2tF4rzxW9ylwqF08tg7bWGRnHzAD9KzNNM0Hhm3EmrqoOpffWNW3fIvy4zWp4kdlk8VbtSiIGz9y0SkJ8wx7n/AOvXqVnfGK2zk+//AD+pGPs406ek3e3d9n/d/r8GaUlufFWjvDC+0aYuwmTOB5cnX5uv+Nc2klsPDOi5t3C/2mw27mO3kdPm5rf8Psx17SP9NRQdJjJgCj958kvI9Me1c1GW/wCET0P/AEtT/wATNuMD1FaUI/7Qk3/J3/6iDz1NuMrS/wDSv7v9z+vvtr3cluLbxzthYhZ035LDcfOP+1/LFOglhHxA8OqI3UtbQYbc2Rw3HXpn17VXuSfsvj39+P8Aj5TsPl/fmnxFv+Fh+G/9MX/j2t+w54bj+lTCOlX/AAT/APTFIpzly35uq/m7v+7/AF6alXQfjF4Q0yyitb27uVkjREkZIGIyuc4FZt78VvDr6s01rq0xtjIX2PbPnlySMn2NdH4UvPBKadbzXraMLiOJBL56qrZVefvcE8nnvWTqd78P31d22aexEjcpbvj5pGHUHGOP1B6GvKy2NN5jNxi4ysrvo/69Tx8yjS+r3nRnJcz0Svr31vp2OC8WfE6DXvD2oaVDbSpHOqJG7XOchWU5K7f9njnjJ619XfDn/klnhT/sDWf/AKISvlzxxe+E5tK1pNLt7FbghTBLErh9wdd3P3SCC3p0Ptn6j+HP/JLPCn/YGs//AEQldrjGMpWjbV/PzPdpv9zCysrLTtoeQ/HLwmPFPj5wDKZLXSbVo4ok3GQtLdcZ7fd61w6fCHRlmjWee/IZhuWKINwODk7uCf0PFdX+0DpGq6r8Qov7HcoYtLtRIwlVPvTXIUckdxXEx+A/EcZthfeI57a4cAeWsTSeXg9CwOOK6MO173M018tDz8wc1GChV5Ne17nWP8DfC0emwymTU3kkhV9yYIz0PH17VdOmQ6P8NrnTrJS8NsCieYecfa2/+vVD/hWvjU6VAIvH8zW/lAojRMoRfT73vWhLp9xYfDB7bUpnu54U8t5lQrvIujkn68fka+dwspTxWHcqqn++W3TV6fL+rndTfNGacr+52Xbfdb+qLFsH/wCErhxEh/4kK9f+uIqlah/7K8IZiQf6c/I/66JVmBIj4qh3RSn/AIkKdM/88RVG0jiGmeEMRTA/b35Of+eiV9PG14ekf/TVfzMeWPI9+vTzj/f/AK/ByTJKvh/xO6xR4/tFCDkZ++1aN1NcJ4nv0ACquhsQAeh8oVkXEcR0HxRmKUn+0U5Gf77VpywI/iy/CW87j+wmBXnJBiHH1omlzVL9pf8Apuh5jtFQVrrbp5y/v/1+VjSItTntfC8sdp5qBpDM/ZR5mQT+A/nSajBdDw9qJvrO1jxqqsg8wHePn/2uvPT9KqWSJs8JKLe7QGSTcpZiceZjH6/yqpqSQHwzrCrBcDGtqGDEnPEmMc+9TGP+1XX866f9Pqnmbt0VHlab0+W0f73b8fvOkurFm1zWnaOzTzNPMZ3yY/hHJ+bge+B1qta2AgvPCmUtXaOJ8eVKGH32PHPI96rakkJ8R+IE8icBdKbAYnrtHOM0zRobVm8Kb7SdisEmxkRjt/ev1O7vz271x0ov6km39hf+o8/M3lKh7Tl5db9v73+Lv5f5Drm4I0GcTW+ntt1IALLJgKNjcnJBz/jVi9nMGteJXb7Gyi24Vjz/AAfexziqccGnXPh++E2kzsg1QKUTeT9089evX2q1e21hJ4k8SxvZXAZrNS7Bz84+ToO1dUnBSlFp7y/9Lo+ZrTjgqkryvfT9fP13JLa8R/EHhh1gtSrWx5V/u/e6evSqelaqtt4VtWH2aSM6gwHmADAKqfzBNaFtp2mRat4dK2t6kkcB8ovwAPm4b9axbex0+TwpZrBY3pB1LKq3zEsVXnjHHT8qyw6oNRSva6/9LrvubVfqlGo277dPku/ob73sjXPiPMtkVSJNqSSRoB8ycseo+p46Vbg1KI32nBrfTgXtVbCzqWUbf4eeR71gSaeDrnipf7OuJhJbx7Ytjfvf3idMEH9a1bLTbMajpUsmlXEcqWaqD5b7Y/l+6Tv6/n9a4sVCh9Wi320t/wBeI+Z20J0pydr3Xf8AruYzXunyaNpUtrpunpG2quAnmZVWxHyPm6//AFqNVe1uYvEarFZQbAgcx8b/AJwct83X8qlubXSbbQtKC6fdRxf2sTGuwrhsJ1y54OP0pLz7AreKQmktN5ZiDr+8BkO4Zz83Y+lezeKrXV/i/wDc8PM0eKoU6jV3/Wv5ItaVawnxZps0LW6hdFAUeYN2PJbtu5Hvj8aoWkNw2laIfPt5Ct8xLB1wQWB4w/8AjW3pFtaPr+mTf2bLHKdIVQwjcqq+U3y53Yz25GfeqFhp+mNpOjCO21GNUu3KIiEndvHLfNwK8v20XU32jHpfpX8yJYmnKcVyt3Xl+r8/wNyGGWGx13bZWWTcBgrNw/7xuW5696BqjR621ubTTgsenNIWeQBm+UnHI+761l38EEeh+J2tLW+mke6QOjhk3HzT9w5P9O1UV0y4PiOV5NOunzozpuDY5MTfL06msqVKE/a1JPpL/wBN0fMb9mk3F2+7/M1lCappfh+5MViu24nYR71IA3j7uRz0rG1DRpBouveZHaYlv43xHIDwDJyR68/5xUMMKDQPCCiyuB/plwFQycqd46/Lzn8KfNY6WdN1+Ka31V0fUogREqh2f94flB7df0r08PB06skn9t/+pL8xKpRhHWaV+un37voat/pwbxJfzukRQaQiHLfM37odBUnh+SNI/C4sIIIoHeQEOdzj96c4P1zSajZ6bJ4ivWa31RpBpCBzGE2+WYsADvu4+maq6L/ZEM3hFIYdSyZJDCZABjMp+/g+v14rzkufArd2j/7gqeZPtIJfxF+H+fyLGqXM7aHfPbRwLIL3YNhU7jz2Knn8K0F+2nXteDGMILDCcx5DbE6jbn88iuavVsJND1XZDqPGp4dc/wAeTwvz9Mf/AKq1J7SybWfFEiR3izNpZWR3AKAbE6YJOeB27Gtq1OMac1/i6edAU5t+8qit/Xn/AFYntbK+mvvDk3lW7pGcyO5RWHPYYB7dvyrNi0G4/wCED8kxQxtJqiyjEyMuAuM5zirWj6RayTeGp4odQfyQSj+UdgGerc8VNbaZpTeETCNE1DyBqKsYdsm4uFxvwTnH6Voqns3BR6Sh0/6eVfMweHjVfPLW/l3JtQtrs3OtM0MRRrRAgDLyQUyDjkfjUcZdtU8L7bWJj9hYs2fucngdf51JdafpxuNcaTR78mS0QM+HIm+5wOeuQOnoaj+z28WteG44tNvDEtiVQfMPJ+Y8MM9frmvOjJfVYr+7/wC6/qYYijeaT8ns+/qv618jIsLe5h8L6cYoLTJvC+x2HIMY5+tXtfjMFx4wlW3hcs0T7j1PzDAPPt7VWht9Pl8MWw/4R7UYFXUSfKV2JRtq/OSSOMVZ1saWureK/tNneniNp2U8ONw+78309Ohr0alXmxtrPSXl/wA/qT7/ANeZyfUVSpqPMne+tvJ95f1+KXQFlfXtHkEEIRdIQMwHzAlJMAe3Fc5GH/4RLQ/3UY/4mbfzFb2lC2bxhobW8E0SjShsUkkBfLl4PPWucijiHhPQ8RTZ/tNuufUV0Uf94X/bn/uwcCjG0rvv0/wf3/6/PUuQ32Xx5+7X/j5XHv8AvzUsYkHxD8N/uY/+PWDPt8rc/wBagukT7L49+STm5XPv+/PSpra2jl+Ifh3bBOdlpA/QnGAeT7Z5+lTC1qt/5J/+mKRXs4yjZX3XT+8/7/8AX4LF8O/BrwzrMMFxffbF+0Ijna5Xlhk4PI69OKy9R+EmhW2qPbwi6EayMod0YjG8gHIYdhyce9XdF+Gni25gM9t40ey3FGEQhJKgE7e/BBzWHqXw51WbUSlz4pa8naVkLfZlbLb2Vv4/UsT9a8zAc7x04uopKy92y0ObHT5aOlfk1etr/L5GX4n+Glvo/h3VNXgadIbURtCsgJDhyuMHvjc35e/H1j8Of+SWeFP+wNZ/+iEr468Q+EvE2maXd3Oo3JktIl5PnoQ67goIXOcZK9uMivsX4c/8ks8Kf9gaz/8ARCV1yvd3aer2tp5aHpr+HDW+i17+Z418efE+r+GfiIraIiObrSLeOZWQNlfNufXp1NcNZfEHxt5cEn/CNfac4Kt5JCtz2AH4cV6H8b/FcXhfx7L5omVrzSLVEkhK5TbNdHoQcg5riJfiZ4avrq2nuCYwoAlDxszAZJIGOP8A9ddGFbg5yikm+t1r+DPOzKnGpCnGdL2iu/l5mz/wn3xDXR4seA4hCihAq79xGeDsznHHXFaH268vPhe9zqkR0+5lTdJAc/KftTev4fnT4vjb4Jj06C3Fzfb44FjMn2f5uD09KlbULfW/hvPf2LxeTcgyK8ilePtTYyMfWvncPdYnD89Lk/fL56vXbqdtK3JNONlyLsum2qa08/mJBKo8VQ5udv8AxIU7f9MRVG0lU6X4Q/0nd/p78Y/6aJxWrbxN/wAJRCVe3J/sNRg7v+eI56dKp2sLf2X4SBktsLfPgjd837xOnHWvpIyjeHpH/wBNVzK0OR/PrDvH+7/XrYqXEqjQfFH+k4/4mKcY/wBtq1vtSxeM7xvtfyjRCWYLnAEQ5/CqM8Lf2D4mAktsHUEJJLfL87dfl61rXNxdLr1/AJrZE/sUnaqncD5Qwd23OPxoqNOVReUv/TdDyKiqfKm/LrDvLy/r7w0+9tPJ8Og6ljfLLs3KOfnPPT2HpWHqN79o0DWXaYJjWIskAYB2yc9Pb9K0rETOvhIvJayEySjLbjv+ftleP0qCW6uxoWrMLi0LLq0aruU7cYk4Py1dGMViW1/Mu3/P+fkat0ZRSm7Ly5P5V5f16WL92baTxPr3makBv04q3HRdq89P84qDTriKC88JxQarIImil6JhZB5r8nn/ADitG+ubhNc1uELYoIdOZx8rFgdo5JxyOenvVGwurk3XhRDJbSGWNwwAPP7x+ny9O1efR1wS7ci7f9A8/I3qfVoVN9b/AN3+b/P+rkMmpQ23hu+kj1g4/tMjckA64Jxj+tLeXfleJfFci3e147JcZUDBGzAqjdXl6fB14/7oN/avlhhwcbW46Vqy3M0fibxO7R27rFaA4deo+TrgZNdsoRSm7dZdv56PkczlhnNcr3t/L2l5Pz/q5oWV8j33hsHV4ZXlg+dGUZfg8jj6/lWRZXNtH4XtMahEqf2jjO5WxwvT3/xrci1OQ6poUZWxi+1W5coA5Yg7sbeMD8cVzdvJLc+D4HgtreErqfZm+XCLyPfHFcmEprkhdW95dv56/kdGLq0ozXvJ2TvrHy7xf6fka1zqEC6h4klh1JY5o4UWSRoxlFDp+eckCrFpqatqGjxHVGxNYh2j2R8nb1z159uKpahPd+Z4taT7PsjRVQeh3R9cDpzU+nT3S6posUdpAYZdKGHG75OCfz471hVpL6qnbRR8v+gdeRFHEU/bT5d/+3f5vJf166mKJbd/C+iFL7cp1ZyQVVSSAvy/L+H51oXr6dGvilzqjP5jJ5yrGMr8/wBef0qpptvdt4d0e2FvAZP7UZgHfG0BEPGe/J4rRvft+PEeYLMYZPKaRgQwMme/sf5V6Fed8Qox/m7r/oIj5eROFULSco3/APAX9l+S9NfyLukyWsfiexSPVchdGTy02j5gI2+brnoc1Thv7WO20g22sIyveMFlKqN5LLkDj3q5pjXEXiixMi2hVdEB2g87tjHjA6dvpUaNcS2mk5tdMd5JmdgjnCjKklePvV8/Jcrk32h/L2reR2wcHWi0rKz/AJfLyv8A1r0tBfagbbS/Fs1prQlkjuYzkrnyDvI28jHt+FSwzQR+I5ZH1j520nIA54EbEuR7dat31nPLpXiTZBp+6SVSdrMQwDnO/wCXg4HbNQXIvf7cDrYW7AaU4Eyn7zeU2FxxxzW9NxqUpwj1jL+X/n3R8jOdnK3RNdVbr5f133KltNDD4f8AC5h11grXcvKoR5+XGRjtis95LWDTNYZNUCBdXRpZNv8AqzmXp68AVqRNfXen+GJnt4fOW4m3qCQFXeoyMfhVDUftTeHNdBgtQ8moADDHLD95yc9+OK9Wi0sRZveo+3/QRLyOLFqEqPurSz6x25fRr+u2hqXtzGvijU421ARqmkIUUgcjygc5/HNZ+krH9u8HA6jFIMtgqw/eES9B9DxVm+uLuLxRqKxsoWPRFYZP3W8peRUOjajdT3/hGS6NuzyNIGYjB/1pHy4Fc9KLWDuv5P8A3BU8jNrDp+89b94fzLy7/wBXMW5ki/4RnWd96Ygurk5VckYDdOn0rakaA+JfFG67ct/ZB3DYDsGyPnr+nvUEt89roGqTIYHC6oynzdzdQwx93it9pzNqWvgW1llNOxvCkyN8inDfLyPz7VviKj5J6ae9+dDyN4U6L0uvvj5+X9feZul6qEk8KxRam5Fxw0Xlr84yevp+FU4tZVvA/nyXry/8TVEZ/LVcfLnGBT8SjxJ4REkUCHaciMFQPmPQYrN09X/4QJmdIdqa3GzBslcbO+Oa3jTg3CX96Pb/AJ+1fIhYp0/cjKy9Y9l5ef8ASNXVNXki1HxJHb34dYLGN1BiXCHMfOe/XvVi31e4n1Lwkk97hZ7AyOQo+Y5Yf0FU9RjLat4reQ2+1bSMMgDcZaM9cdOK0Le9WS/8KRJYWweSyZ1kLEmMZbKj24rz6kYfU6fu/Z8v+gY6I4hzquM6ml/5k/teUfl+G+pmWF1DbeFIRBrYkH9pbVdkCb/lX5e/TrVjWp7Zdc8X/a7piipFuWPG5cMMD8ODWcl+JvCNiFtrW2xqm08Ft2FXngdea0NeBfUPGWDBgLGDkNkfODzxXVOly4xykt5Pt/z+orsZSrU3blaaS7xttLvH+vS9zTbiOTxtoAhmYxNpO5VcDccpJycVz8cqnwnoeLrP/EzbjHuK3NFD/wDCYeHsqmP7GGef+mclY8Yl/wCES0PKQ4/tNu/uK6KSSxEV/g/92Dlcqbi369Yf3f7v9fda9cyL9l8e/vulyn/Af35prThfH3hrbMZSYbVQFO3uQenpzx7VJch/svjv5Y/+PlMe/wC/PWiQlPHnhszJBgwWg5BPJPGPf3rKNuWr/gn/AOo9IqDp3XqusP5n/d/r11Oc0n4g+PtPtoodO8FSXEUQVIpNkp3AZAzjrWVf+M/GMutfbJvCM9vOzYEXmzKpO4nGzOOvt2rsNB+M/g/SLKC3u57rzIFVWWOA7Qy5zjJrndQ+JnhifxCL2K6eSJZvMQyWjs6jeWIyX9+OOK8zCW+vSc6NlZe9pr96OfFRm6L5KSbu9P107nE+IPHHiXU9HvtO1Kzjhgm2rNiEKw2EYBOM8YFfYfw5/wCSWeFP+wNZ/wDohK+TvFXxD03WtE1Wzt1u2e6VI4md1CKqsp5XHX5Tznvg5wK+sfhz/wAks8Kf9gaz/wDRCV2SSUnZW1fz8z0o/wAOOltFp28jyb41waJP8QmGum3yNKs/J8+QpjM11uIwRkjjg1zc2j+BFvLRYvsLptjLMLvZ8oIbIBU7vxPIyK6j4yeDrbxd8RZVu7iaFbLRrWXEIGWzNdZ6/TpXmkvwq0SHUUhfVrlUd1Cp5W5ivIySGwCe3HHTmt8JUXNOMXd2110R5Wa+zUKbqVZQ1duW+v3Hpc2g+DF0O0ka00ZY/s2dwmTeWOerbeR0Gfes+6gt7b4d3MFpCDBGCsaRNuVR9qPfvn174NYkvwM8Nx6Vb3X9rXp8y3EhZVBDMc8j246frWqunQaN8MZtOtppPJt1KI56sPtTcn8/1r5rBypvF0PZ1HL98t76as9Om1yytJv3F/N2376+WpYgC/8ACVw5t3b/AIkKdP8AriKo2m3+zPCH+jSD/T35P/XROa0LdgPFUOZ3X/iQp0H/AExHtVG0Zf7L8If6Q5/09+Mf9NE9q+qi9Yeke/8Az6rmfNLlej6/z94/1/wLkdwF/sHxR/o0h/4mKcj/AH2rQutv/CU6hi3df+JC3X/riOKo3DL/AGD4ozcOP+JinGP9tvatK62P4qvgkshP9hNkt0/1Q9qcn71T0l3/AOfdAOaXItH0/n7y/r+kU9PC+X4N/wBEkP72X8f3nSqVxt/4RvWv9Hcj+2Y+B34lrW06FfI8Is2oKpjlkO3PLZf+HjmqZBj8P62UvzDjV4/3gB9JOOn+cVrTmliJf4l3/wCf8ypRnaN0/wDyf+Vf1/wS/qG3/hJ/EeLSUf8AEqPXt8o/z+NVNN2/b/Bn+jyH91JwO372Tite8tkk8R68xvZT5mnFR+7/ANke3tVaxjsre98JJJc3RnEcgjOwKuPMc5OQT17cV51CovqSX9xd/wDoHmb1KdT2l2tL95/zf1/wxiXcWzwVdFrSVd+rblJ7ja3/ANetS98v+3vF+62kc/Yxyp4/gp180B8JXCzalOkaagAA6biuQwxnaOMEmrd3DZnX/E4+3spe1CyAKTtHy+1dkql1K66y7/z0SI05Odr7ec+z/r/g2K9sIv7f8IgWc6P9jBBJGB97kj/69ZOnqD4Lg/0OV8aq2cDp8i+/+cV0dklp/bHhlm1KRm+yYjAT74GevHTn9Ky4o7OPwrarZ6tKsL6mdzvESW+VcjgVnQqe7Ba7x7/z1/maVqM25NNdes/L+v6Y2/KjUvGmbeQD7Ony/wDbRKmsgv8AwkPh3/RZP+QSP/QTz/n0qzcwWR1DxYz6jcu0luokWK3G5PnToT1z/Wp7WGzXXNDaO5vy66cFQNCoXbtP3j2PPT3rCrUX1TZ/D5/9A8RRo1PaN+feX81/T+uxlaTJZ2fhrSFuNKmneTU3EeXxsYrH82AeR/hVy/jEi+JkTS7gspjfZj74D5z1/wA4pum3Fva+HdIWO8u41bUWUZWINnCfKcjp9OeatX4tfL8RyNf3TFwizYVTtAYcDj3706tR/XVvrL+9/wA/4/I3hSqewfX5y7d93/XUsWMQXxBpZOkPFINH2tMzcKPJb5OvUY9KzbLyF8PaO93pN0gF2VBjIIjO9cMTkcVqWchufFWm77u6ZBo28q0Y2kmJhnI6Ng5rPs0trfw/okVvfXwjN0W2iNdxbcODxwK4XHnaT39z+b/p/wD12FTnUhJu1/8AwLsv6/pk80+nLp/iwjS7tsXSeau4DzD5jD5fYEZ/Gh304a9PvsbtJf7HYlhtwieUeBz1xU84tV0nxMf7RvDuuFaXEQJjJkbhfl5z+NE7Wv8Aa26TU5Wf+ypP3Rt1yq+WTknHp2opN3nGz0Uv5v8An3RLSk48zsu1+bu/x/raxShSCLR/CyfZL3DXExjR1G4DevLc9x/Os7VYIF8N+I4xZXIkF/G53KBuOZORz061t28sN1o3h24k1W4dxPMYn+zAtL8w4PHygevFUbpbWfTNeF/f3gT7bGBIY1QrzL8oGDkcnmumnUn7dN7+0be+/wBYl5fmEcNPldra+c+yX9f5jtTT/iqtR3W8jbtAXDD/AK5Af/Wqt4fVftngzNpIxDy4PH7v98eTWxqEdkfEuob7qVT/AGOq48wYKeUO2049c/pVfRhp0F34TEd5NndIIgGDiQmXofkHf6UqVT/YrW+x/wC4KhyVKFS6u+vef8y/r/gXOcvtn/CLazm2kI/tjkjv96tqUx/8JN4q2QSA/wBjncSQd37uPpUuow6WPD+o+XqRaNr/AHSfu1+R+eDlP8frVwmzGueJS07bl04hwIwcL5ac/d5+nNdFarenPR/a796BUqM4NJv8Z+f9f0jL0+SKPVvBy/2eQXhGGkJyhz94YPf3rPtEMnw7m2WUhMmtJsRBnGF5P0rct5LJtU8JeVfSOxX5A1uDuHpnA21GYoYvCBis9TlCHVU+dofKwSmSvA5+uKtVkpwVteaL69KtUiVOo433S85dkUtSULqPjDNpKubOIgkdt0fH+fSrVkYv7T8HBrSR2/s04Ib7vLdqLq3gfW/FX2i+YrJYoQFPI5j5ORjOQPzqxZxQ/wBqeEj9pZSunHb84O7lvbmuSdRPCQX93z/6BghGftnp1/v/AM33f12OUiC/8IfY/uH/AOQy3H/AE5rc17b/AGt43zbux/d/MP8AeWsgqsXhe0QXJcLrTDcO/wAi8dBXS67BaC98YM93dRO5TeDECowRjb6816WJmliov+8+/wDz/pGNKM2npsu8+z/r/g2KeiY/4THw7+5Yf8SYc+n7uTisWML/AMInof8Ao0g/4mbc/iK3dGKf8Jl4eCSuR/Y3AYYz+7k5NYkbL/wieh/6Q5/4mbcY9x7VNP8A3iPpD/3YJ5pcr07/AM/93+v6ZoXOPsvj39y//Hymf9r9+ajnx/wnnhnELxny7T5m7fN/X+tS3JH2Xx7+9b/j5TPH3f35qO4YHx54ZxK0h8u04Ydfm/p/Sso/DV/wT7/9A9IqMpaaPdfz/wAz/r/gWLPhHSvBd1a28l/a6RJMqx+YJI8EtjvngnJ5xxWfq+m/D2PWnVksY2D8osT4G6RgOhxj5cDjjg9CKy9C+Cej63DFc3Mt8onVJPlfGd3LY+U/hWbf/B+xtdUe1t4754xIyiVmOMbiAchcdBz9fSvKy2UP7QnyVHKVl7rvZHmZjKi6Fp15RV3qn57bdCp44tPBsWh6xJpX2Q3zbWgEUjbk+ddw25xyCT04x+f1D8Of+SWeFP8AsDWf/ohK+UfFfw60fR/Dd/f2N7dvNaxRS+VKBjDsoH1+8efavq74c/8AJLPCn/YGs/8A0QldjkpNu7er3/L0PZjb2ULa6LXv6+Z4r+0BoWp638QEOlSpCLfSrUyvJKsY+aa5AGWIzzXCr4B8do0Sf2kw2RrsO5CF5BCg57ZJ/Cu5/aD1rU9I8fKmlxLMtzpNsk6PGHBUS3OOoPqa89f4heLJ5rcy6HbzeWIxta3Vt4BGO3GSR09sV14a9pc23S1/xODHrFcsPq3LvrzfodVcfDr4jW+lQtH4vaWIJ+6gRiNq56AHgf8A1614LO+sPhdLaX06T3cKbZpG53N9qbp+v6Vz0nxE+IcukwLdeEI5YXg2pM9uf3gGfm9OpHbHFb9tc3F/8KWudQsxFcSR7nhUYEZ+1NxjtnA/KvnKHt/rWH9ry/xla3q9/Pv5nevaKnL2lvh/G2u9lbsXrcT/APCVQ7TF/wAgFeo/6YiqVqJ/7L8IbjDj7c/QHp5iVct0U+Koc2+7/iQr/wCiRVO3WP8As3wgY7VkH25+rZ/5aJzX0kfip+kf/TVfzOW8eR7deke8f739fddJI530TxMqmH5tSjHIP99q353vYdbuSv2RtukuABuDFvJGM84xWN5H2zR/EyLYZIvwR5YwSQznPNad5YkalfyrYSgtozfPs+XHkgbc+uaiq06k0/73b/n3Q8wu1FONradI95f3v6++zbaXUkh8J+Y1qRNM+5cZ53HB/DNVoobyXRtXR5bNni1hQhuFLLxvOP1/nT7MW4PhPy9O3DdIB8wwvz4Pb1P60268iHw5q0sulAxyaimAAPmceZluR9B+NVD+O0l9pdv+f8/MVSt7sNV90dPdX99f15amxJLdN4s1aFPsazJYMdxVx1UdecY6dKzrSSZLvwiP9FlMiSqHO5ip8xuQc/T9aW+i2eLNdKWOW/solVYAjoPz/wDr1FYxtJqHg95NNUERTblRQqr87cYHT1/GvMpcv1OEunIu3/QPU8zvqyvN3S37R/mX97+t99CuLm5h8H3zItkg/tEq26AtwQeSD1PTmtu4v73+2tfhi/s8slqrx+ZFtwOB8xzg9f5VhWscH/CM3Yn0zz1fUwAnHPynp/nvWteWCP4m8Sv/AGbIyyWqqd3CN9zvXdVdP32+8u389HzMaWImqijG17LpHs/736fra9a3t7Hq+hWdw2mrJJblmCxMW/i+4RwBx3qvZ2+qx6BbpF/ZYdbwlyYGCspA55789ahNoieLvC5h03yFWBwUJztHzcVR0+y3eFFQ6HOzR6kT5ZIJA+X5uR0/wrlwyjKlTnG2rjfb+ev5/qzuWKiq8qcknbba/T+95+X+erdzXUs3iSJl05hHEmFIY5O5PvfN/h2p9q11D4h0eIRWAY2AORG3yjHQHPSqmoG0jm8Sk6OwAt1MrsVPm4dOAMDge57VftVtZb7THg0Eg/YVdJsDEaleFzUVbLCLT7Pl/wA+I+YRqKdRxvG/otr+v9MzUvbiDR7ArcaVJJc6o4doy4UcJnAzwR6dOlR6jbOYvEW67sXEjR8LKflO8fe54rPtBLL4b0dhpsCD+2G3KkQ4GI+ef88VoXdvsHigvpYwChHCfON309u9dtZRhiL3WkvL/n/E1pYqlKny9Onw22vorv8AryLun2rL4ssJzfWjFdF2eUrksf3Z+Ye39Ko2VrJLouigalp0Y+2bzmVl3fMDtHPJ9vWrulpC3iTT2+zxrnRAVYIAQvlMPTOOgrLs7S2k8OaD5OnRyKL8kS/MNvzZyPy78cVz0G3UV30j27VxRxVOEW0l/wCS+Xn5/wBaGpeyalDpPihmWyH+kxiHaeCpkPLc/wCeakfTr1tZMjTaaEOkuFDnkEoR3/h/pUF55UGi+KR/YkK7JlLluROTK3zHJ7DmoJYbRdalZ7GRGbRnVZQ+Rt8kkjGff9axpTbVTTpLtr+7o+ZlLER0u09t1Hz80XLay1NdH8PKl3YO0c83nYXKuCwxt46iqGo6XqdtpHiBmvtPYzX0bozyALGAXyOejcj9al0+0h/sLwwn9myTL9ondRvI8r5xzxWdrH2NfDviBW0Qxf8AExjBdZmzKcud3I7envXXg5OeIlyvTn7L/oIfmaTrxhSTqSV/lv8Afb8TV1J7uPxVqUSyQMq6KrAnnBEQGfzq5pMuqLJ4chaKzZRJIsxCvkBZcZTn055qneRQP4pvkNm7mTREG4MM/wCqAwOPfFR6RgX/AIWT+zWRg8uAwU+Tic57Vx0/ewVrfY8v+fFTzOGpiHGWlRL5Re7XeXy/4JFrWsasPCupSSNYkm/MSbF5VMt1x34rVQX0moeIVje3ZDYYT5TvV/LTg88isjULeK28P36T6Yrs+qj76KA+TnoPb+daU93af8JB4ltjpkSNHppZ5gQWcbE+QjH09eldFW3s5xiv5n070PMiGIjU/eOStp0jd79eb+vygsYtTW+8KvPJbqoAWRdpyST2pLGw1GLwexhksZZRqnm7pEZU27cHg45qOySIXfhErpuwkx4kHROvH4/0rNWBj8P7lTYsuNXjYLnoNv3vyqpJtwg7ayh0X/P2qu5NLEwcpSkk7bfDp7qf839empt6hBex6jr8omsWla0iMahGBX50PJ6Yzn36VXtJLw6v4RAkUg6e3mZJwTl6sTQ29xdeJY10kGVYFRmz/rB+7O3GR0/pT7OyMV74ZZtM8xE04qJM/wCpOScdT6+9czmlhIJ/y+X/AEDeokrVXJW36Jfzf4v6fnoc3La3h8JafJIbfLakZiVBxhlBx9etdHfSahLrPi37KtiWiZNm8YY4wcN7VhGxA8G6WZLJVVNQaRxG4+YbQA38gfpVzWYRNrXjVUtiWPl/Nnr8wzXfK0qyXaUu3/P+ku5Eajp3btey6R/lb/m/r01LWnwXD+NNDnl8lMaThkRGwD5b9DjGPxrnQky+FdEAeBlGptgqDzyPWt/SbVk8VeH5NvyrpITknOTFIaxEsXj8J6MJLCZHXUi3CZ4yOT6VVCUXXg79If8AuwNpOMuWK69I/wB3+9/X3Xt3Il+y+PMeX/x8pt4/6bnrUdyJR478M+b5ZXy7POwc/e4/pT7lR9l8e/uetymf9r9+ajuFVfHnhnbD5X7u0+b0+b+v9aUfhq/4J/8AqPSIi1pot10j/M/739euhzGi+A/G2oNE9h4yjsxII3hQzkMAclRgcjAz+tZmo/Dvxm2qGC511LiWSRkDtOMucsGwCcn1/GtPR/HPxFso0j0vw211BF5ax5i3DaOF6c84/wAKpal4/wDiFLdESaVJaPub92MZBLse47cr9FrgwHt/rkva25LK1vi+dtTLGvGKinRcb3e97eXz7nLeI/B/iHSbW/l1C9SaO3UNLiVSWUsFHGc9SPpxX2N8Of8AklnhT/sDWf8A6ISvjrxF4v8AEl/Y6nZ6hYpbw3TA3AESggqRjLYz/CPrX2L8Of8AklnhT/sDWf8A6ISuupzcz5v12+Z3xv7OPNvZX9fI8k+Nviez8N/EPN/DcypcaXabTbMAQVmujg5ByCDXFj4x6GJ1k+x6iih9wRXK4GcgHB59z3rvfjBB4bn+Isw8UGIY0a1+y+bK0Y3ma5BOQeQOCR+VYlmnhxIrSPTp9MFuv3TkHgNzyykn8SDWmDvzVHTi7218/TVHjZxGg6dP28HJX0tdW9bFVvjv4ZNnGgstW8xYhGeeD15+9yferx1W31n4aXGpQGWKG6BkQP8Aex9rbr15/wAa6wT+HJtKgldtIWVrddwzH6+nXisO/wDKk8A3X2aaCOAklCpJUD7ScdP6V81hHSeJoOEHH98r3v3PTpQjGM1GNvcXXyfr+RWt2QeKoczOP+JCnQf9MR7VStGj/svwhiZz/p78Ef8ATRPataCFl8TQOLqDnQ1AGDn/AFI5+lVLGINpfhUPdwkLeSFSFb5v3icDj+dfTqSvD0j/AOmq/kZ2ag7/AJrvH+7/AF+Tbe6jt9J8Rt9rlTbqSkgHb1ZhjOD29q15ZoBrt3vv5tv9jsdgJ+VfLGT93GfxqkyI2ka+bW6iiLagm4ndxh26/wD1q0J5J5vFl5ZC4ttj6W/z7SNpMa9/TjNY1WuepLyl/wCkUPIu9oxhF3enVd5f3f6/PLgMENx4UWyup2h82UbipyV8zvx/nFQa3NF/wjupCK9ndP7ShLlgRtYiXIHHTgflW/a2d3A+gf8AExsSIzIWAz8+WyAvHbpUF9pkt7o9+r6lYlheRhSFbCbVkwD7/N+la0asfrF3/Mv/AE/PyJlhsRNxVtv7y/lS/l7/ANdCW8jsm1/VJV1C4Ex04q48s8D1Bx2ArN06UQXPhON7qZWaOXIKn5v3rcnjrgmumme8XVr7Zf2a/wCiEgMp+UgcMfaqSJdXF54enlv7e4KQuWePcA/znkfp1968LD1f9nUZbcq/9R6n909SWGctb63fVfzJ9vl+G+pgrHazeF33ao1qy6kGL+Wx2sA3GAvpWzdPYnWtf8zVLsP9nAkwoKxjCcrxmqEr3sXh28EjQq323AZgzDbhvfOasyyBdd8QpEW3C3BYlW2jO3/a/lXpVLyhUV+su381HyLpYflnGX5u21/IvW0mnNqmhltTu3k8kiEEYEg+bk/L1/KsUvaL4dtxa6xePG2p4DnJbG0ZUnH+c1rWs0x1XQlldfmhORg89fespJZv+EcthbCKQnUsfOGUdF6YbrWODi1Sppt7x7fz1vI66lGMpvRfhf8AIdrBt1XxUBfXLM9srFSflT506Db3zV22ubVdY0pE1S8Eh01THDztk+XqRt6/lTjBcS6l4haVljDRIsbMrFcblycbj6egq4l06a5psbzW5kNqMN5Jyfl7HP6VNWX+zOK19z/3BHyOf6o1UdXbp+N+39fictYzJH4d04vqEwD6s+1lfljiPgjHPU1d1C9VrfxSJdSnZYnQKuPujcOmfyqdpkl0fTllnR8ak65ED8nCH6jr1qxJaSTDxBGktrmaVWbeSOcjk5rur1Ie3Skt5f8AueHkctPDVFB2e3n5Nfy/1+BDpTxjxPp2J5ht0DKIy4z8h68enP1rI06VP7E8PoL6cFr+UnC8Odw4PtzXbW810utW8UlxYug07LIMbyRH25+7/hWZZPcHTdMkjk0tS9w5OcYwTyF56nFcOGrP2mq/l6+VfyInh4uOs+n+X93y/H7sO6jmn0nxkk08jZu4ypIZto81uBgVbkihl8QFPtzKRorKyurcZiYE/d7cVr29zeyp4jVZ9JhdZ0Eci4OBvYfvMHrxx71BPe3cWtODfW7KNLdyFgGdwQ4OeuO+PStHVcZTilbSX/puj5eRawlOcd3pZ7rz/u+ZmxzNBo3hOGHUpFia6mWQ5I8zDjjGOcVWlktbnT/EUV5eTSRLqS/w7ip3ScAEegrQluJ20/wy880TTNdTEM9rksdw6Y4U+9Zsk2oNYeI1tJY0mOpIOI+x8wHORg/hW+AXtI3va8t/+5l+RtVwk/aqzva+l12t/L8/x8jY1K7hj8SXcEdxMhTRwxOwcfu1xg4znHNZugzRSXHg55Lq688ySkHqJMzHhu9aWpR38WuX7Rltj6SoPAxvEQ/wqHwx9te18LeVN5UCvL5ok2hn/enjH4Vy05Qp5e5X05V+NCp5GVbCVZVE3rptdd7/AMvRef8AmUbv7LeeH74TarJCyaoGZpEbCkE8DCn8K0pbGL/hJPE1x9uBE2mkFPLcsg2pz93B6ds9auTz38Om3ZFxbRM15hGVNwAyeoBOOlSDUdTXxD4gijdcQ2IeDIGC21eRntmodWbjOS2afVdXQ029DWWFw9JKnPdf8HXb1MyyW1k1LwztvpnZMFFKkbgCeSCox29O9ZVq5fwPc7Lif5dXQKxUknCdOnStyzGovqHh2e4uoDIQTN++O5yTzgDg1V06O8fwi26/hjkOpr85mfAGz7uc9fbpXX7SKcJN396P/pyq+xzPB0It+80vltZL+XsR3m2K88VSm7lQm3RlIU9P3fOcdcj9adZTmPUPCYnmuUEmmkd8FtzfeyK2kub5dQ1eOPUrEiOFNiOGbyz8nLdvXp7VdS9uZJdIDahYnzYNzLsIMhyeVz0FcDxF8NGKX2f/AHWt2NVhcP7XWpZ72vHvft30/wCCeey3sf8AwimmPBfTBjqWxyeMEouVyO3etu/S3TxD4wa7uJY4mCElFPHI4+7/ACzV67kvn0OKSW+0xm+3ffRWClcDgejVHqup3a6j4rjF9DDHBt8vOWMeSByOn5V6FSu6tdxgtVJrfvWovt5mCw+HppRjNtW/u9mu1u5T0+VD490PyZ5GRtIBXd/EPLk56dayHvJm8L6L5mpTyO2oFThjyMjC9K6bSXl/4SbQ/M1aByNMDtAI2y/yP8wOMY9s9qjm1C7XRbGYXthdH7bsLx59cYG49aqnPkrwVr2UP/c/kKEE6cvf37NeXZfp+hiXLJ9l8e/vG/4+Uzx0/fmo7gq3jzwyEkZz5dpww/2qv3Uqtb+NiXh+WdAf3J4/fHrzzTWUP498PEyxMfItiE2Y4z25rSMvdq/4J/8Apil5HPGKurPquq/mf93+vxOV0D4zeHtDijguNP1HfEER2iKkMV4JwfWsfUPixoV3qz3aaHcYMrOpafGcyFhkbff17ke9eseGX8PLp8C3DaYr7E37jHlTznOTWPqbaNHqpTGmPH5znIZOhkJHO/p+BrzcvVCOYz5YNSstbNp/I83MXS9h71JySk9Lter3PLvFHxJ03WvCV5ptvZXcU91FGrs2PLJVlIIGMgAKQPrz0r6v+HP/ACSzwp/2BrP/ANEJXy34ytfCC+H9Vm0trI3jIhh8mdy+8uPMBQ8dMkY7V9SfDn/klnhT/sDWf/ohK6+WMXLlTWr3/pnsxd6ULKystDx348eGB4j8eMxldTaaVaMI4497SbprrgfMOcKf/rVxo+DmmB7dVe8uA+BLIGCY57Lz+Wa6n9oHTtav/iHD/YMjRsmmWokZZQnWa5CjJI7g1xdv4U+KEXkLFqDIdoCEyKSvIOCevGc8+hrswsVJSUnf0tdfeebmVSpTjB06sYa/a2Z2S/A3wi+iQyBr8SSRrKzeZmQZOMYxjH4Zq42nw6L8M5LG0t0mgtQYo2cgkgXR6+561zcnhr4xJbop8Rh1EQZVWQHjJ4+714zz61sxw3tr8LpYdbmaS+RSJ3U/eb7W2f6frXzeHc3icOp1ef8AfLborvTTsehTlLlm+a/urb030d9fU0bWX/irIN1lE4/sRGPAGf3IOM+lR6bKItL8L+bpduTLduVLFRsG9eRz15HHtSW5hHiiIMZd39hJgq3AHkjtVbT3ik0jwmrzTDF6+BuABxImBz/SvouW7hp0j3/59VvMmDqW69eku8f739feaUsZfTte8vSY5d18m1cou/5m54PNXHeOLxFqMT6ShI0sgsXXOfLHyfTFYzy3FrpXiIGW6JXUE8tY2boWbOMVfnaD/hKtV3S3HOkErknkeWMn68VhWg/3npLv/JQ8y5Vqnu2i+nSXeX97+vyuW80Mv/CPtDodnGkpfaskqBoxu6pzye/HrTXxHompO+kWjj+0FJSOeNUIw3zE88/XnmslJIXuPB5WSYqJJACWJON9JGkZ8NarHDe3UWNWDM6Ke4fgDI/P6VrTp2q6/wAy7/8AP+fmdMMVWlZRi7vyfZPa51Uvk/2peodFRn+yklvPQGQY6eo9MmoFt1a+0LZojL5cbkbblCIvmbg/3vXj1qrfSI2u6osV/ckDTmzFvPyt6j04pml+VFcaBHJd3rAWrFfmPzsZHzu/pXz/AL1PCJr+WLt73/PiSfU65V5yfLyvrrZ9Hp1/r8CtqK7NDvpToaqkd6pMXmg4+U/MdrD8ven31zFNq/iWIaajGO03M24jzB8vqxH6Cor6W0TwnfyJcX7O95xktndhsD/d4ov54V1jxQzzXJZbQZjUlVH3eh9a9bDxbou976rr/PRXV/qc31yu6jik0l3i+t3/ADen9bT2Ni8+peG510hVVLYt5ouM+UPmOMZ5rL0+1tR4Qhzp5KjUs/JdgbW2rzkj9Kv2lxB/bvhbyLu5RHtDiN5WO7Ibrxg89/aqWjGA+F7ZpNTuzt1M5eMN/cXj6d811w5kl/iXf+ev5mc6029E9uz7L+8i7dXMKXniuNdOlzHAhY+avzYdOmV4x+PSprK5iOt6GjaazGTTg6szp8o2n/Z68dsfSoLqPbqfisvPdmJrVSu0nIO9cbaLXI13w+RLebf7NxJuzgHbxn3rjnFPDv8Awef/AEDx8xqvWctb79n3/wAXb+raE1hsOh6bL/YNi8b6g7LGZ1IQYTDAnq3H6VPfyR2ya4ZtBgODH1kX99lxgn8a5aBUm0DQw9xJPnWHw2W/ux8ciuhurLTnk8Rea14Uk8vcB7MOmT61vjKSWKhe/wAT2v8A8/4eZrSrzqUpRindeT6p+f6mpbSBPEdmG0i2iY6Yf3m9SUxFnYPUc49OazrK5i/sfTGfQ7T/AI+3ACuoWLDAZHvyOlWLG2s18R2Unmy+aukbFQkkEeV1J9cZrB0+Cxbw/oZaS4OL6QgDP3t4wD7cVx4dR9tbX7Pf+Wv5nPWq1YU7u+nk+6/vf1+e1bXPnW3ikJolrJsvETYNo88+YeW6dOv41KUm/teU/wBkQfLpbZfanUofkznOO3pWXY3Km18YCe4uVjS+jyyOcr+9bp6VINr+KpNs9wFfScE8njyWyfrWkotVKq6KLfX/AJ90fMUcbVhHZ6v+V935vsWZYrmDT/DSSaf5bfapg0SKhVMsMcg8fhWbdxTHTvEYis2jc6hGPmQNnmTkZOP/ANdbkcmmHSdBMl1ebTPL5DMfmZtw4bP6VjS3NhHp+tvHcOv/ABMk81wHDKcyeh56e1Xls29ovSfZ/wDQQ/M7pVJ6Tk1rrqnfbyen9dTU1NboeILqRbNXjXSQN7IoDN5a8dag0W+tZY/CYfSFEryMFlWb/UsJcE4PUE8+1XNVuoItXv8A95M0kelq2NzAMPKBz6A1DpN5pLzeF8fbd8skghAJIBEnO/2zXPT1wOsX8P8A7gqeZj7SpKVlNL1T797/ACM+S/t5dJ1AxaEZv+Jp5XlLMP3jc89OPpWs8jJrniPOkMALDKyZHz/Inyf0/Cs26+x2uhX7Pd3hb+0CQwUoQ3PTrx15rXZ7V9Y1rfc3eP7Oy43HC/KvK+/T9arFNKDSWl3370PMwnXrRtN66dE77S397yX9NWxba5t21vwpu0vy2kQ7CGH7v5uvH0qzpEDT+CP9G0jz/wDiYBvLluEfcMfezwOOmOtQafLbHWvCub242tG2xTk7zk4zTFuoIvARaxvL5wdURWcyfN7jjtjtWmIVnShFP44/zf8APysu5cJSlGU3pouj8unN/X56eQdT18f2EkgWFMt5ijzvufLn2x+lX1SN9Q0MnRFR1tSQfNTEA3H5ff8ACshfsw1zxGGupkLW6EhHO4L8mDgdD/jV/wA2Bb7w+pa5Ja0OD5x6ZPLAjnmvNdJ/V4xaesV3/wCgf1HKeI9peK080+/r2/z2KsmntcaBbQ2/h9UIvtwQzo2MAfPnP4fhUOtuLa+8VSS6XDKvyMCXGZRuA5FQwXGkJ4atQb68MP2/aoLkkNsX5ODyKm8Qy6Z9q8UxC5vPNwjTqi8Idw4UE8104WMo4qUGn8b6S/5+0e7ZvONSVKM1G0rdn59L9/MfpLW7eKtHkbThFI+kDZiX/VDY/GD1+vvWNaT2q+F9K8vS02PqDBfPmUlSCOc45P5Vp6Ndxt4w0SIzTMr6MCC5IY/I/UZIzgVmGLRx4a0lILi5eNdRyrs4U5z749u1elCK+sRTT+x3/wCojzOOEazi3FbeT8vP1H3xR7Xx15dps/foD8+fMPnnn2qC4Ujx34ZIg8v93Z/NnOPm/r/Wpr02/wBi8eeW0gIuU3kvnnzj044qCWVR498M+TLISYrQEbuo3c/1qoX5av8Agn/6YpHPy1OZaPddJfzP+9/XoZ+hfA/w1rNnDc3NxqIlmRZWEcigHdnpkVhX/wALPDkGuJptvNchzKyEXF0gfAcrkKBzwM1Z0fQvijfRouleKIIIWCGGN5wCFOSvG3sKp33h34l/2sEuPEVvLcbmxIso6hiD/DnORnp3rgwTqSxs4qopKy92yuvx6nJi3KNK7qqOr1uvu+Rg+JvhtDo2i6pqEFxMyWIQhZYtu7eV24O454brj+E9OK+t/hz/AMks8Kf9gaz/APRCV8eeJfDfjKwsL+bWrmSW1iwZmM4ZWywAIGfUjtx7V9h/Dn/klnhT/sDWf/ohK6p/E9b6vtp5aHoR/hQd76LXv5nj3xz8Sap4d+IobSLOG7NxpFssqSweYABNckEDseetcanxJ8SSiH7R4YjmcBCQZFG7B44OSOSK7j42+Mj4Q8fylVbfe6PaxB1ONoE10T+efb255rg2+KujzXUU/mTQ/MHeNd3UHPPy/N7nvgGtcJ7spyaSut77nl5pRVWFNey9pZvfp5nQP8RvGL6ZbiTwE7RmIhJWl+V19RxjrjpV5Ly4vvhlPd3UK2M8y73hPO0/am4H+e4qrJ8bvCr6LBaeZfedFCELeXgE85Oevp261aGoRax8M59QtFMsVwpdS4wf+Ppv54P5V85hYSjisPzUuT98ut76769zvjGKhN8qXueXbbXTQs27MPFUOJkX/iQr1H/TEe9O0maSHRfC6fabBhJdOT5y8j514XB68/ypturHxVDiEN/xIV7/APTEVVsnkj0nwkFgQBr58kgH/lolfSuKk4Lyj2/59VzGnKELy069Id4/1/wbG9LcXcul6zNc32nxgXe1WCb8KGYYZc9ef50Xlw8+u3cQmt8ppLkOqKhyYxzktyOe4wPWsGeWdtC8S4RV26igDDC4y7c8VtvAx16+E8ULRNo5G37TtI/dDPHTB9cVhOko1Jt22l2/koHbTqUnBRglbS/wd5dvPt/mVWDIfC7vewZSRvLEKKdxLdsE4HHWqt1qs7eHtVZL9x5WpxxqVtVG0Yk+XHf6+1amiwiOy8P+ZbW07J5iowmU4BkwMHjPYVM97bDQ9TI062n23oBSOYZY4fnr1GDxW1OUfrDVr2ku3/P6Z306mHjSu4wj812Xn2/Az7u6YeJNadr2QP8A2c2R5OQg2jkc89BVjR9QuJD4cA1UossD7t1uP3uJHABycjH9Ku3OpR2+t6tGdNQtb2JkeQ87xt6dcd/Sk065a/8A7CnttNQoYXySQPKy7jp17evevGxUqay9ScUvdgrvl/58ytubQqYZzSio312tfz/HfzGN9tbw/Pv8S2aYusiUxKQow3yYz1qzeXt//aWtrJrNtHBFCCgeFD5R+XkjqR9aiWK8/sifzdHtiXu/ljBC5GD833utGppOtx4i8zTwyNACrrglhhcjGaww1e6UNLXf8v8APS2sa1mqblKNvw/UfHNctqWgFdV02YNES48tVaQYblPb2FYuni/uPC8P/E7sUL37bnVRtI2j5eg56/nWxplvPPNoMi2CrFFbguWO1k+9jjP6YrOsrOebw3ELewttxv2IV2RQg2gZHPXvXsUpxVoq2jj2/nrHHJw3STvv8Pb1LV9FdtJ4hxrtpueBAq+WAYcOvU49yPxplvZTDXNFk/ta0kWGw2BAMb8ryR7VDfaPOl34puPKgZbm3UBWkUY+dDzg5A4PpU9pppXWNFmFrERFpwTKtn+E9Pm9/euaU4rDtKS+Dsv+fESNHLWmlb/D33/X18zL07Tp4dF0phqNjG8WqySKVJZSCsY2j3/xFaWrG5lsvEO3VrMyEoNoIXy/mXgkn0rM0yO2PhfTSlpbsBqrECQfcOEzwW69O9bl7b4/trGj/OxjY7ZARKd4x346c1042SWNhJ7qT6L/AJ/w7mFKFCdGajGO2vw9n20+/p5EmnGca1axPqNs0Y0oP5SjLSN5RG4c9O9Y8Rnj0HQYo9StWRtQfcQANw3H3/zmtuzaYeILQf2aEU6TvMhb7n7sjb19eKyLe7nstF0U3dirtLesm7cPkyQQePTIrjw3+86W2j2/lrHFOGHdK0rJekf7v62/qw/TLqZF8WsLq3tT9tTbK0WQuZWHIzVx7qaLXJljuYpUfTGYkRDIYRHB+90x/wDrptnJd39n4oEVjAzC5Ecf7wDeRI3JBOBjj61aY3EOtTlrC1aP+ziSZGAwfLb5evQ454orO9art8L7f8+6R10adN0kna1+0e7/AK/4Nxq3FxHpGgn7bZyM106s3kg5G4cr83GO55rl5J7h9L8RhHiydTQcqQAMyZzg+1dFLqRhsfDcb6Rb7rq6lUJv4i+Ycrj1rN1PV7n+xdZk/s+yYR6giose75hmQZbB68fzrfKLpaR0c9NtvrD7GmLWHlJ801pfpHt5+Wv4bGnqcUr6xexi/tkDaQB5RBzkxAbic9OKh0KK5tj4WH220fEkok4JaTMv8PNXtQlm/ta+kXTRKi6aGDgkbj5QO3g89ccCs+1vGkn8KD+zDAXuDuJZlMX70joTzn3rGg3LBf8Abnl/z4qESo4GOspK9+0d7/52/wCHIL/VLuz0fU2GqWqMupFVby2YIpB7Z6/pWuL2RtV1mNtUtMR6eGEbRnMXyodzc8jnpWYzPNouomHT/PIvyMC9ALdec4GPpWwuDrGsn7D92xwzG4GHO1fl9vr7VOOcI022ur/l70Drp0cPblUla3Zef/B+4xtMhS61Twq8mo2xaONyoEOPM+boOeOvpRbQRweEJoY9ZtI0XVBmUwqF+593GTyfWtGwhe4vtCuYrFBFg4ZLjcE54PXnpVe0hQeEWM+mROq3ys0ZmUDO373Xg89PetFWUqsYt7Tjp7u/tKz/AAIjSw8ZNe76tR2svz0/qxNLBd/2prcser2aFrWMK5WPMf3Mluc4x6+oqeJbuPUfD3/Ezt2VbQrJgIRL8x+YfNnHA6ZrNv712v8AxDDb6XaRtHaKpf5S0r5jxuz2+b9KupHcPqHhsLYWUO20JeMum+PlvlUdcd8isJpvC05aL3fL/oHHz0ZT5VrZ9o9/Tv8A1cpz3c6eHYn/ALTspM3+3dDDww2rxgtwe+as65AUk8TSy6raQo+AFKbmjAIPzYOeTxiuetrWWx0WygOnRzP/AGuR5cUgkUfInJzuH8q09atr+81PxdCLfcj7BCzYA+8M813OhGnjG09HK/T/AJ+0V+hn7WElZQTdrfZ8/wDhv6Zc0TD6xoN1JqEZlOlqPJ8nJICv827PH0rB1W/nutE0h31CCffqRDNFCEBwR2zWppqtD4w0SJ44M/2SqkhslcRv09q51k2eGNFCRq6jVGw+Rz0rooQisVGX+Dt/1EHnSqR9i4qyVtfh3vHz/r7i9cu/2Px4vmpt+1KRx0PnHrUV0SfHfhndKjjy7PhRj+L+n9KkuQfsvjz90P8Aj5THP3v35qO6BHjvwzmIJ+7s+Qenzf1/rTj8NX/BP/1HpGMeXTbdfy/zP+v+CYmgeMfHGnxwi18GTX9ugTySQOQBheQM4PUfjRe+PPHL3DbvCt1bvlsgSJkfMT3T8Pw9an0P4z+GtJsIrW7+3b7eNIyqRIykoMcZA6n3qhf/ABp0SfUXlit7wxmRiMtGMguSONhPTHfuRXn5dFRx05Onyqy97v8AocWZUqtSguWgp6vS/wCPzOT8W+NvEeqaXqFhqekR2sMuFkYw/Om1gcb8+oGcAZr63+HP/JLPCn/YGs//AEQlfJXib4lQ61oGoabGswS6ijjVS/yLsZCCB6/Kw7cEehz9a/Dn/klnhT/sDWf/AKISunl5W9Lav5+Z7UdKcVa2i07eR5P8aTog+IEn9upbMx0m0W3NwX2qTNdZbCkZxxxXMSweGnksU00WawkKS8MMTLjdjkuCe/fnArf+OfhCbxb8QSIJUj+xaRayPuUn5TNc5P3h0AJrjIvhBokRtorqa7nkkbDyw5RDz1AIJH4mt8NKC51d+er0+5HmZlKnBU3Um46u1uv3npqWvg+TSbeSS28PpcNCu/5oTzn1H484rF1BLZfAF19lEYtsHyhbEbQv2pvT8P1rNHwJ8KSWUNzHLqASSEPjzwTuJ9NvTpV19Kt9D+GM1hZTkQwAopkIJIF03+P6187hHTeJw/s5uX75b308jupNctTlk78q79n2X+Y+38v/AISqHPm/8gFOhP8AzxFVLGNZNO8ILGsxb7dIepP8afpV23bHiqH/AEhV/wCJCvX/AK4iqdlLs0zwjmdH/wBNkAHTH7xPQ19Or3hbtH/01XM05cr1fX+bvH+7/X32uS2Fs+i+JI5Fuo2e+UgornJ3NjqRx9OBWu0FoNZuzcLfJnSioKhyCvljOPmxu/DNUvtLLaa9JPqwtw98ih8BvLG9sDG7ir16Nmp390NbRiNKYrAqgkfu/vgZ/GuWo5c803/N3/ko+R6NJvk5rfjLu+lkQ6bDpUdtoKm21JyGkEDRhwqfPnL5P86oXAt4NE1NoNGvcjVELDzJVLHEnzrg5x9OOakt9QZV8L7dQjk8wtljGoLHzBnv/nFUbq9SLw5rD/2gCV1OL5hEDtJEnH3uf/rV00Yy+sttv4l1f/P+ZUKzsuWP4Psu8f6/A2b29j/tbV4TFdArYM5PnSYI29AM4zz2pNNmgf8A4RvMd4paJ8APJj77/e5/n60moalt1PWQt9G7JprNsGQ4O36/y5qvpfiG7H/CMRNeqPtMTbwUB3jzJAec+wryauHnWwCS092PV9KMn2O516kKiv8Ak/TsSXl9p0fh+4kiTUgi3wU+bPIGzhuRhsge1W76SxfUfECyi4OLceZ+9fBGFxxuAH4YrKXWi3hqd5tQiBOoiIu0KMF4PXLc/Wr+o6qY9Y8Qql9EFit8kfZ0YoflweuW/Grlh5Rq8sb6OXV7c9HyKpyqVYuXffS35ou6dLp6XmixA3nmfZiYgszBf4vvLvOfxzWTZzaa3hy1+0Le7JL8gK10zndtH8S449vrTrTUy3iDw4BdQMZrck4twC/LdCDx9Ko2mqNL4XtZXv7YN/aBG77MmDwvGMjB9+tdtOjJctu66v8AnreQRozc2uZm/MtlNNr8clreNujUORO5Ei7kwAAeDnrioorawXX9Jb7Nel1sQkZEuFUbehycniqdxcxJc+JmkvY32wxkx+WuEG9Op3c/pVq01KP+1tJjF3D81krBfJXJ+XqDu4HtXBUpTVLRv4H3/wCfEfI39k2tf6+5GTHpkcmgaXHa2l5Fs1R22l9xHEfJ9sCtK+iiH/CQPDc6jC7tHk7X+T5x9055/CqUeqImi6VLFcxyF9UdSygKM4TjG/3/APrVZl1qAf8ACRie6AW3ZQSLcNgFh/tc816GIjVliYvopPv/AM/4eRjHDQSkldN+b/4H9eZpWECN4ktJDPfM39kgYYPtI8s8tk/e/XNZdpZ20mh6Mssl+3lXxKvskPG7gEE9OnPTirOnamh8V6fBDLmJ9G3BSmMjyiQfvHH0/Wsm11aH+ydAb7T8kt8VPyNz8w6Dd/OuPD06vt767R/Kt5HNLDQlG02//Jv66f1qakVpY29j4pfffSCS8RpdkTIw/enhTnkdeRipJYtFbXJiZL5Lj+ymHQlfL8tuTn+LGcVWS9kOn+LVNxDB5d4i73kICgyMMn5uD+VSf6K3iKVX1CPc2kNhQ7cjy2yx+bFPllerzN3tL/03R8h/V0neO2m/N/X9ehKsWlSaf4Z3NeFVuJmt9zNuY7h1yc/rVb7CP7K1pIptTjjbUFbMTEleXztx2yefwprx2J0nwuWv1k23U3lSBj8x3DODu7fWm3X2H+yNbz4gCL9vj3v5kv7o5f5eD9enHFb4OLi7Xfxvv/0ESOj2NPm95tel/T+v8zW1Fo/+EhvYHm1DP9mLjbNgAeWOQufve/rVXTX0tpvCwL3Uknmv5O+UnJ83+L5jnnPrUmqRWT+ILzfrCxyf2SAV3PlF8sfP1x7+tVNK+zW1z4QS21iK5DSyKCVbMoMhzgn345rkp2WC3fwef/Pip5Gc6VPl95yf3mfrMax+HdVwt44OpbgXuAnHPQngDPFbC3Vquv8AiBWjuwY9My+27G0jYn3Rn5T7/WsfVngvvDuqi91aKUJqOF8x5CF6/L1yP5Vspb2H9v8AiHdqSI7aZiRAW/dLsT5vvewPGOta47ldFqV95bX/AOnAQ9hBuTu9OvN/l6/1Yo6TdxR33haFIbgBgQP9OG0DP8Sj71Q2E9g/gc/aobqKMaqqsI70MSdnBLHtjHyj0qWwFouqeFMagHZt20KXO/n/AHv55qO3j0//AIQebZqw2LqyESNv5bGNv3s9OfSm5U4VoWTu5x/m6VKv9fkbNYScLtOyt/N1/pf1cs3k+jjU/ER+y3LSCxRpyLg4cZjxjHQ9Pyqa1urB9U8LMbW6WWXTyYisxOwbmJDHvT2fbq2u/Z9WtEcWqY3Egx8x/MecYI/nUk15KniDwxA97FKXtHZnU5V+W+Yc9+lYy/3aCV/h7v8A6BjhqqNOopL8VLq/T+vQ5q3CXHh20nEd3bxf2wW+VndvuKM5+tbesxWp1Pxdg3gdgvmEbto5H3cdaqaVqM8/hm1uG1q3tSup7Pu/eBVfk9zWxqs97Fd+J867BAgZfJRm3GAHAOfT/wCvXbXqT+u8rVrN9/8An9T8jGkoyhfe+/xdvlb+upmaUsY8daBtExRdIUZYkHHlydc9+KznW0m8K6QzfbI9t+NrSktk8fkDxW9pKyS+KNDabU4ZD/ZoJXGfNOx/mz6VnG/li8O6X9mvbKdTfeXuYAAjA6Bj1ohUf1qCXaHf/p/5GEE4KSls7/zv+Xsv6+8qXOPsvj7h/wDj5T+I/wDPc9KVfJPj/wAOeZBPJ/o9ttAkxtbJwx4OQPSnXcha28eZmTi4QcADb++PWmyK03j3w2FnjfbDaMRkDIB/z+VaRfu1f8E//UekYQlK6t3X8/8AM/L+vSxr+GrXwdHYwG/g0ZXCJlZxGZFbkENnv2NY2p/8IsmteQ1hpRQzMBJD5QULvOCef6Vn6J8EfDGrWaXV3JqHmSqsjCOZQPmz0JXmsu7+EXhiDUDapNehvNZcNIOFDkZ4HoPSvMy+cI5jUcKr5rLR7fLQ8nMXQdB+0nKyk9r3/DoY/jFPB6+GtX/s77B/aGUNv9nLhh8y7wRkr/exjtX1H8Of+SWeFP8AsDWf/ohK+RvE/wANBoWjahqEd7HMlqquY/LYMoZlC5O4jkNnp27cV9c/Dn/klnhT/sDWf/ohK7JcrlLld9X/AMMe1Br2ULO6ste55B8cNP8AE1/8Rx/wic7wSR6Ram4ZbnycqZrkAE5Gee1cDDpXxWtpI0imLNgbZCVkJ5B5Y5z1zyexrsf2hddvNG8diOzginS70m2WVZE3cCa5IxyMda89m+JOvXlxbGfQopSqxjYUJ8zB4/M4/GunCKUXKT07Wv8AicOPp1qkYKlCMtdebX7jqzp3xnh0+JFvrfyljGyJFQnbk8cL7ZrdgfU4/hc8esxwy38aYuNygDd9qYdvp+Yrl3+J3iyfR7RW8GSNGlvsjmCPsdf7wGP5HtXQW15JqPwpa7u7RreWaPc0CA/u/wDSm4/lj8a+eoxrLFYdVYxX75fD6vfXfueglNU5c0Uvd/T5adjTt5WPiiIC1gP/ABI1Occ/6kcfSqVrK/8AZfhIm1twGvn6L0/eJyKmtwn/AAlUOY5T/wASFOmf+eIqnZrF/ZXhImOYEXzkemfMT9K+iSV4ekf/AE1X8zku+R6d+nnH+9/X56cs0sek69J9ktMpqCbQQfm+Y4z71oXMGof2xeT+TbpEdIO0s4GG8vpyemRWRJa2sGj6894Lvyn1NdzCNuztwOfwq9qKW/8AbOoNBay/Noz/ALxi2QPK4GM46VjNL2k7f3v/AEih5m1KNqab/D1l/ef9fhHp5u5rfw0fKiI3vvKNx/rR/n86qWrzW2j65N9lhld9SXbG7eiyc4/CrGgtCmmeG82187M0gBjJCD94fvUyRre50HWRDoN1dSDUlDxtO3zHEmGyOw9PeuinpiZK2nOv/T8/M0jSc0mpWk9tH/Kv739fiaN1c3B13WY2t7YounbgAw352jr+fpS2E9w58PAW8WBvDksuQfMkGBx04p8gRNY1Rv7GuixsyHdZDh8L90c1WiVjqnhp49Luo4/Kl3Aycx/vHwDzz6/jXjSg6mCShb4Y/hRlfqej9XqwqKbd129X/i/T/IkuFvNR8O3RtbG1ieO8DgBQc4BOenWrt7Z6hPea4GtoWjkhXyycDdwvX5c+vrWRfBU0C42WV7FGt4N65cs3ytk/T6VZv5RHqviKXyrrbFbjcRuA6KQBzSrUbtcttJu3/gdHzNqdNxuqlRarX8fP5/Jlm2t746xoUjW0IQQESEKOD85GML/hWXZ6fcL4etVksEkkXU3IXcQAdoGfu/8A1verFg1x/bnh9JLacP8AZC7fO3yjLdQfw5NVoWjl8NWjyreH/TjnYGfnavp2rqoqScFpvHa/81fzM1Bc8+WSur9PJef9fnp3VrLbyeIp7W1iEjQr8ykFmO9c/wAJ9PfpSReYNf0UyWFuZjYDErltw+Q5GMAVBchLaTxNPJHdhREpJWNhj506HPJ+mO9MX7PL4o0LyorpmewyrMWGAUPOK44xbpNvX3Za6/8APiPmOb5tG1fT039d/wCvIZ/xM/7C0vzNN0+NjqbEjyui4TBAHRjz+QqfUBdpD4iMkGmoodNpdnXOWH3jn8sVQgt4X0PS0htdUYLqbkF2bcTtj+Y9ML7exqPVvLltfFQjjui6eWHVnJAxJ16cV7U1fEJf3v8A3PDzM03SptqXT77J+f8AX4mzp11IPE2nQG204r/ZAbzFJLA+S3C5P3f6Vl2U7LouhvJpWkyBrxhgcLH86jcvzdeaTSR/xWelfK//ACAB/wCiGrJtIydA8MqiTZbUmGOefnWohTXtV6Q/Kv5nL9aqWb5l93mvPzOmN7I+neKEttP02J4rtAOSVlJkbJfnr/Wnvb30niM/6Jb+W2lOp2lsFjH0xnGKxoRANO8bm6hmliF5GWVOp/et06VqLdW0fiWVI4LoAaKZCCTggRnH8XpXNUXs3Usukv8A03Q8y4V6k9HL8PXzEtoL4aL4bjjtrVSt1OJVGcKN4HHNZt/JP/Y+vSmO12nUUUIZjtYDzAc88Gti3thPpfhyQWN3GI7qdiioxKjcOW+bjPvmoX0xBo+sxzaXqExmv1d44lZS3zPhlz1GMZrahVjCs7reo/8A1Ik+5tL6zUjydPTy9e5fnmY+IbyIx2uxNLRyC53YMQz9AMenejTT/pPhpY7SBkkeTEgn+5iTPyjHNS6laGTXbwjTrsq2mBFkQN8x8sfL9e3HNQaXYWiHwvvsdSWWB22fKxCEyHO854GfXtXmQnBYNX/l/wDcFTzFOjWaShFerXn6mTrViW8O6wz2VuiPqG8Yul5685xweelbHkFNU1yf7NCV/s0jcJvmf5UOCMcfWsuS2gn0PUVuNKvI4/7SbcpDuX6jcBuHf0rens4GvNbEVlfebJYFd+xtrAoo2rzyfw7GtcXZ02nfSTfXvQ8yKdGtKLem22vn5+hl6LYXDP4ZnFpCViB3vv5XOeR+f6VD9juB4NktTZ2vnLqKE/OSOEyW+vXj3q3p32C0uPDsFxbXyXIG2PzGI2n/AGhu/wAaZanRD4Rkeztb8xNqClgpJZpNvXO7pWUJVfrfNJXipRa0/wCnlT+96mtTBTnQtT0lp3/K67dyOedodS8TRrBagi1V/nJPIKA59un5CtGJLuW98PuVs0AtD5kYUZBy33T6VSntIP7Q1+Y6ZqOyS1QNIoOJOY+F56/4Gia3CeJPC7pbXkcYsmTa6HKjLHDHPWtJ2lhaaX8n/ut6nNTpVKUuaWt359Zb6y7f1bQpWS3yaGsf2OOWQamdwaTouE5H59PStfUrG8e+8SCO2t3E7r5ZkuCu7kdf7v8AWsPTobWbQbcS2moSo2qOwADlh8i/TjpWhrkVk1x4rIS8eQlDIfLcIDkdCG5/DFdVdP61GC3T/wDc1LzM8OkpznJadtunqaWm2l5beJdJFxZQIF0/YzCZmKsFb5eeo9656QXS+GtNhdbeCQajy7SEYGc4zjryBWhoqWx8SeH5wtwW/snGQjlSNr/xE9evvWE6258L6X+4vM/bs/vM5DfL156VVKL+sx/7c/8Ac/mc9ecFdqDt6+ce0l/X42rm4uRbeOfm5S4XZ83/AE2NRXNxcL448Nqx2I0dpu2t1y39abchfsvj35JP+PlM+/789KjuAv8Awnnhnaki/u7Tl+n3q1ily1dPsT/9R6RnFy09V3/mf97+vwOb0aL4rywg6Rq1pDbqoMfmFMhcEj+EnpVO+j+KA1Ux3Op2RuMtlkaIDhiD29R+tXdF+IfjPTIYkt/Bc17DHtEUnkMCygYX5gOeOh9+9Ub7xx44uLxp/wDhFbiGbcSCbQEqd5bumeh2/hnrXHgY1vrknUglG2jS97563MMbGt7L93CDd3vt/wAP3Oe8UaP46s9Gvf7evzLaRqrTxfbd+FLAKdm7pu9uK+u/hz/ySzwp/wBgaz/9EJXx14j8a61qNrqdnf6fHb/amHn5ixIhBU4yTkcquRgdK+xfhz/ySzwp/wBgaz/9EJW84yUnzLr/AF8z0I83s4829ltt8jyP43eJrTw58QT9tWYi50q0AMWOizXRIOSODn9K5eL4o+GDJFKuoz2q7gzQrFnHOcE+Wcnnkg9hiux+MNj4dvfiLKPE0scfl6Pam2DzGMM5musjII9vpXJTaH4ZM1jFp1rZGNwp3iETKV3Y5ctz1Hqa2wu8+SLv1/4Gp5ObU8POnSjiE7Xdrd/M23+Mvg5tNhhGsXayRQBOLc4LZ652dfwx7U2XUbfVfhzc39nOTBcZdHK4Lj7W3OMD+Q6106+HfCz6BBE2m6VgxKzRbk2bsnnA4z15rM1a1iHw/kj0m1jWBYwiRouAALg9McY618xglh/rNB0ouP75Xv3vqd1JQUZ8qfwLttbQo27D/hKof9I2/wDEhXt/0xFUrRgdL8If6Rn/AE9+Mf8ATROK07e3uh4oiYW4x/YarlgevkjiqdnBKmn+EVuokiAvZOuR/GuO1fVKSvD0j/6armdlyvbr1j3j5f19xYmuP+KZ10nU8OupA7WQNs/eH1H+cVoz3l3ca7e2MV/G0P8AYpYq6KMMU4OcZqGe4+yaLrUkj2N5Gl6uV3ucZduG469uKuTyySa9dKljagjSWKyfvM8xjjPTHT3rlck5Tdv5u38lHyOqjRjTSXP8rx6uTtovP8PUpWKfZ7bwzDFq1o4V33YRTv8Anz8pIzkH0qzOI10PU2fxDOD9uXLxwDMZw/yAdx9fSqtvJJIvhgNHBvMzkff4IfGOnTnvU76lexaVrLQR6bmK/RFJhJTbh+vHJyB+tbQTeIf+Jdv+f8/I2VXD35J2t01XZehYnkRde1ZpNRP/AB4k/MmVA2jnBGM+1SQXUNzcaDKdYimmMbEZgGZBvYEgkcY6dulSTapdrrWoQCLT9sNg0ylo254H3uOR7CnWmqXc7aERFpbx3CSBmQMMEM33Mjp069815dNWwysvsLt/z4n5HaqlC9lNavy3v+j0MiRVuPDlzBdaysKteFfMRcheGwOg61Y1JXN54li/tJSjWy4iC/cOF68Zqe9l1NNAupE0/TZHiufkUMSCuDzk9TzTrq5uHvNdtvKstyxZUs+HOdvU44HNTWUrc0bW5m+n89HyMXGi6rild2t07PyGQlh4g0CSPUklD2zRsTGNzjkjnHTP8qy1We08M25tdYUSNqGXNsnlhvkUbcAAVv27zHUNHUrpwYxHO2RiR977vy4P44rFsP7Rk0GJEu9Kkf7dhmaNgoAReOnDVpgZpqC0+x2/nreRpPDR5pXdr36rsvIn1e4lFp4lEd88rGIKIyB8gMijI47gmo7KSePxJoQW6Yp/Zaq2cABtv+fzq7It/Lc+IE8uxn/dRhB5UgJ+ZeG29e/TnpTY5LuLxBpdvJbW+WsMs6HAX5TwMru7etTpHD8tlflk+nWhDyI+qwpSc3Pdrt3bt8Pnb+rmWDqDaJprS6qqv/arNmNtwYYjwuQOO/5026SWRvGG+7kCuYwpCAlhv6D/AD3qezu9RutB09o7aFB/aLK6P+8G0KnoOvWjU3u/J8UIlrbptMYDIQpclhkk4FexNtYi2nxeX/QRAyqQw8oc3Pf7uz/u9v66Bplu6+MNNZXYouhgHIAyfJasm2inXQvDW2V98eouSABx861s6Za3TeLNLmaNcHRQjEEdfJPGPxrNgsbsaH4dU6euUv2JXcfkG9eaypy/exu1tD/0mucSp05KVrW17d4/3f6/KykUUek+Mzd3DRI92u446L5rY/OtW0Nv/wAJAsn2ry3GlYL7VOB5f0ycf0qBl1G00nxTiwjIa5XylD58wNK2WPpxWgzXKanLCq2EYGllifMO8NsI546dOa4cTJyVS3aXVf8APugdGFpxhO7+58veX91f187se4sRpugibVZHL3UqxO0CnzG3fT5QPUVh6hqVu2ja+Yr+a3Eeoxp5qwLkH5+MDGeQetX4RqE+jeGceRM63shlYvgEB+2ev4VG/hfUf7L1uGa2gnFzfpLEnmlOMvkkgdeRXXQVKnWbm9VPy/6CGuxu69ep7tFWXy7X6L8jQvp7cazdmXVGDDSlZosfdHlD5sY/Gk066SN/C8dtrT7JJH/dhOJxvOQeOPSrd7pdz/a11KtvCEbTViDPIx58sDGNvt1zVe3i1IX3hsOLOKOORhOA3X95xtyMnj9a8f3Z4VJP7PW3/PmfkbzqVIJOW3qu5k3upNdaFqU7aqwkjvduZiFCYJGBgcdK0n1Er4g19DqJxDppcYyfL+RDnp75/Gs6fS7q28PatHPDaYF8Zt0rsFZTk9gOmfzqxMhfxN4iljEALaWfl399sfUdhXqVIwdOaX97t3oHnrG1mk1K+z3Xn5eS/rc0+4SW68Mu2oJKZj0MSgydTnJGR0H51HDNZ2XhGZU1do4/7SVTLBGPvFM7ccda0tO+1efoCfZbEpjc7o5yucn5eOehrPbTbubw/cGWK0WQ6sshjAYpgR7f7vXvXDQmnjeVvS66r/n7PyOqGKrulzR3VtLrte239bj7t7X+0deRtWlX/Q0MgMYIjG5BkHv/APXq0t3arrHh6IXuTJZExv5S4kG5vbj9Kp36Xk2o+IbeGK1ZzZoVC5yfnTg8Yx1qRbG6TXfCryJAjR6eRIhJOD82cce9XaLwdNyf2PL/AKB/Qw+s1XXcWlo+6/mt2/r1K+nXcLeHrQ2+tIkY1BgreXg42j5OV68/rVzWBp0s3iiOfVfKy6+ehi/1fIxyBk//AF6ytNhubXwzB5c2mLINSJbuuNqnA46kCtHUbGWS78YFY7OVZHUnzJGBGMHBx09a6qtNRxzlf7Xl/wA/qXkHt04rmUbW7x7Py/pXJfDrxNqmhr9t2bdLVki2glvlfnpx1zWJO8UPh/TVj1ISq1867lVfmJIOOlbelWEkeu6HuubBRb6b5bR5/eElH5GR0/HsazJo54fCumweZp0qNfcyWxJXAPb3rSny/Wo6/wAn/ufyMJ0lLDyul98e8fL+tPIpXJ/0Xx7++xi5Tt939+ajuTu8d+GQJfM/d2fGOvzf0/pVq5iH2Xxz+9i+a4XPytx++PXj+VO8tB468PNMscmLe2wqBsg54bpWkZLlq/4J/wDpikYwjFtarddY/wAz8v69TI8PfF7wno9jFBe3d4ZYERWTyFIygxgHb0znGTWZqPxb8M3GqPNE2oFPMYjCRAMC5Pdc4x75wfWu58NeF/B9xYwSXmlaQWZFYmVE3ZOc7ga5rVtC8PxeIo7b+zLGKAzOD9ngj2bfMOMknI4wOBivKy9YeGZVOWLU7LVrR+h5+PpYapQtODaUpbd+v3nDeKfiLpOseFb/AEyzhmWW6SMFmxtDIVIxzkcAjpz+efqz4c/8ks8Kf9gaz/8ARCV8teM9K8I22g6q+mC3W9jjQw+VclssXUONm7jA3dsY/KvqX4c/8ks8Kf8AYGs//RCV3uMYtqKa1e57aadKFtrKx458efDMXiLx4zz3j2sdnpNqx8uIuzbprocDI6AE1wc3wm061ntIWv72fzzhpEeOMDnGQrc/zrtfj/aa7dfESH/hHTMHXS7bzfKkKf8ALW6x3Geh/WuBj0v4mJLAQrSSxqhjkeASFeRj5iDkjdnr610YeLkpa/lp955+OnUgoONSMFf7Wlzq4vgN4en0yG9i1rUTHNCsgAZGIJOPTke9bDaTaaL8J3063Mtx9mLRiRmBJX7UeuPfP5VhTWHxnt7COT+1oXQINqIilsZP+xz0rXj+3r8LXXW5Qt+qYnZFA3N9qbOeOo4/M14FCVR4nDqdVT/fLbpvvZHZDncJvmT93p6b6a6lm3VT4qhzCx/4kKdP+uIqGwijn0fwmNvlOl85AIPzYdec1PbsP+EqhzO6/wDEhXoP+mI9qo2rL/ZfhD/SHP8Ap78Ef9NE46V9Gk24eke//PqsZJz5Xe73/n7x8v69Lk8lgo0PxKbm3lEb3yuGjwxwHbnrWpdW1omtXUotbozNozjDYCMPK6eueKyJbtjoniITXsxSLUowq4B2je3ABFaWpw2X/CSXbLdySO2hy5HlgAL5Y5zgc96ipz+0nd/zd/8An3QOijUio2qc3l8XeXf+vwKtnEMeET9k27XkLgcBRvz3NTXtnbjwxq0cVm6s+ooTtm6n95zz/nmq9lDbRp4TRLpihaTJABz+8+nv+lLPBp7eH9XX7fJg6ohZtqcNiTC9K1teu7N/Eu//AD/mddNYZxvJz6d+y/T8bmrLFBH4l1WV7N3iOm4fLr8zAdB+FSwwrNceG3ttIuIY0R9xQBljG5uG9M4zketRX1tpS6trDSXdyW+wt5ioq/d29vl6/U1HZXlst94ajttYvo4ZoHwgiCibDv8AexwMHI6dq8iEb4a8bu0Fvf8A58TOvmoxSipPX17/AJX/AAKMun3L+Er6GTSrppZtQBH7s5AwfmA9P8a1n0+1bWfEZa3u5GlhCvtUADhfukdT/hWZLqdzb+Gb65j1+7eaO/2xvg5+6cJgjp/hU93fXs2veKIzqjwpHagx54EZ+TnIHHXrXdWhVmpWdtX3/wCflHyPPpuhCpaLk3583Z9V/XzsbNlbWcV1oYm0ydbhIsI3O1DzkH9ayfI0SHQ7aNrS5iRb4sU34Zn2ryfbp+VSae1wdU8NsdXklzbjeqLuWTluc+nvVKB3g0O23a6yldSIEogAJGwfLgj/ADmsKVNxrJ8z6dZfz1vI63LmV5R0XnO+3p/XrcmvbW0a68UvG10zSQJvSOPO350xt568e1W7a1tP7c0ZhZzErp4VTtOR8v8AP2qlqEFzLeeKCb+Qq0ChDjGz94hH51JE7p4g8PRvfyu40w7wBxnZ9KmrF/V01L7D7/8APiPkZupGMnaL3X8/fzVv+B5GQLK2Xw3pItIbrausOQk0eGBxGDnGR26/WtkWu258Tve6XcvbNImxUjUFwG7E9e3WsaCbHhXR4/7TZlbVm3EoTuOIzt6f5zWrLqawS+KHW9u0MMsYLKy4wXAyBt/POc16WIU3Wsv5n3/6CIHNRmovmk2v/Auz6f1r5luzgsv+EtsDbw3EUsej4O8DaqeScA471kWLWbaD4fY6ddOPtzBTHITtO4ck+lblrqKT+KLGJL+Y79H3mN41w4MRO7IAwa5a3ZR4c8NYvHJ/tFvlx1+dfauSjGTqrm7Q7/y1yZ1JRbcHpr/N3Xr/AF8zoXSxbRvFIOm3YzdDzPnOZT5rYK+2aZMbZfE5j+zyRltDcgsgI+434njtVXzoodL8YN9vkG28QMVX/VZmbgZHNTToreJ2f+0o2xojqquuSB5R+bGOlRye9UTb2l3/AOfdAyjUxDl5afzd5f8AA/qxPaw2o0Hw2z2zuVupRHsjKhSXHUe/vVC92to2uJd6fP8AJqKZMLYZ/mkwee3+NJbXLrovhFFvW2PfThnRcbwHHbHaq9zbNNpPiF0vFdn1NGJZAdv+s4IIxXXRjy4iXM/tvv8A9BMgc6n2b6/4/wCVeT/zv5mvqCTDxRqISCVlOirtO3PzeUKSxt1+2eEglhKphkbcWP8Aq/3uMn8aZqLk+LdRiaVgRoa/LGg3E+UvfFWNCMH/ABTbXGqTRTs7gRMnM/77jJ28c/SvLUH9VjLtDz/58THUdWStG+/Vy/mXl/XoZOrhbnwrqfk2MzM190Vc5OWBI9uKnkTb4n8V7onUnRz1GM/u46muLK0k0PUYm1a4VF1AtuVSpQ5bgZUep/xqzizPinxNuvZkddN2uXGQq7E+YcH2/PpXbOpanNf4u/egV9UrRUXKTu/Ofn5ef9aDNGjgc+FZRp0rYQATAHEZ7k/rVQjT28KzpdWV0kQ1lVWGNArbtvGeefrV+xltheeFY4dWuJFJJVRGAsg9/lB71jJdRz+DriY3kpH9toVkkbBJ2ey469sVzUaT+tRavZOPV9atUvnrQjJN3u/73SKXbv8A1e5sXMenNqniBX025ZjZr5u3q43JwOe39KFtP+Jz4VaPT5ykdiQWKn93y2AecdaZd3kkWoeJfK1eTKWasq7QPLOU5ztxz/WlW7mbXfCbnUZHEmnkuQR+8PzckY9eKxipfU6ev2fP/oHByft7O+/Tm/mXy/rsZkFvNF4Tt1vdEl3DUjxsZcfIvz4/StLWJbZtS8YK9jI+xV3lRjccjnP/ANbtWRba3fT+F4J5tUnEr6qylwoy/wAi/L0962NSmurnV/F8LalNGkYVIl52x8jOOK9CtGSxXvfzPv8A8/qJMpe0jyxcnppfm7S/r7vISzWMeMNA8u0eILprgs3b5JML9RzXMRIh8J6FiBsnU2/mK6m0llj8Z6HCNRaaM6aTsJ5Y+W/zEY7/ANK5yG+mXwno229lJk1JgxIGTyO+K1oc3t427Q7/APUQcsac6cXGTfX+f+7/AF/TNC5tZDbeOv8ARpP3lwpHH3/3x6U1rMt478OP5EkSpDa/ORwCCePxplyR9l8e/vW4uUzx939+aZNIF8eeGdsnm5itBh1B/i+nb+lRHm5av+Cff/nxSLi3db7rrL+Z+X9eljK0n4EaLq8Inm1XUVeQK5EZXb8xPQkdOKzLv4N6DDqBtotavCwlKbJWVW4Yj0PpVnSNL+L11bR/2RqdvDa7UaAMUUbDyuPl7DrWbf6R8VxqZe4kSSfeQZY0TBOTznA7j9RXDgZ1Xjpp1IyjZe7pdHFjnU9jdV1DV63X3arp1Of8SfDiDRtD1K/t9Smn+whN8bwFcbioA3Z54cHgYODX1z8Of+SWeFP+wNZ/+iEr498S6V44ttP1CTXJLiS1j2/aGMpZDkjbxnpnGOK+wvhz/wAks8Kf9gaz/wDRCV0zupO7vq+2nloenG/s4Nu7ste/meM/HvxRc+GviEptrO3uhc6VahlnQttKzXJDDDDBya8/m+LOp3c8DXHh9ZMCPcA0gL4PUYbucevpzXqXxi8Tab4b+IkzapHu+1aNaxxt5Cy7cTXO7gkYODwRXLW/j7wdmJ1vY4FyDt8oZAznn923Prg4yOK2wlnKatbTfueVm3KqdNui6mr2ureehRf4z6rPpVtB/wAIfMWig2iRXk2keoAGccep6Hmt+2v5dV+Fcl8bcWzTpvaFyTs/0psDnn/IrSb4p+Cf7KhhXXEV4oQiqsLfj/yzwPwFUb7U7G9+HV3fW5E1rNlomUHp9rPbH07dq+cwsUsVh7UuT98vnrvr33PSpL2ilCNO3uLz6bdNizbCT/hK4ceX/wAgFev/AFxFUrUS/wBleEM+Xj7c/T/rolPt7q1PiyAGPcToCHoenkCqNpd2h0rwcRFjdqDgHaevmJ+VfTx3h6R/9NVyfqtXkfuPr0XeP97y/rrYnEv/AAj/AIox5eP7Rj6/77VqX4m/4SK63eVj+wnxj/rkKwbi7tP+Ef8AFRMWdupICdp6739q1Ly5tD4nu0SEK39gOT7jyhTn8c35S/8ATdAl4Wryx9x9Oi7y/vef9dJdNmuP+KObEJJkmH3cfx47Csucyr4c1oqUU/2vHg+gxLUmn3Npt8GAwg7ppR06/vOlZ93eWo8L66SnC6zGCNp64l56V0UElin/AIl/6fmKphaso/A3p/Kv5Uv5vl/Vzsrq6nttb1sGK2mX+zBLhwTnaOh/OotOnhmvPCsvl2MUzxSbURJOPnfhe3X1qheX1oviLxAn2aNimkF8lSc/IPz+nvUGn3dsb/weDbw4aJ8YHT96/AryqVJLCba8i/8AUeZ1ulUUrOGl39lfzd+a/wDV/IvzrI/hW7lZbdMaksgVZAdxGfw/Ci7e6h8Q+K3VY1H2Tem5c/3Pz6Vj3N7p7eAb1obKNCuqFXLfMScMc4I644q5f3VmNZ8XBogzraElsHkYj/OutxspX7y7fz0fMweHm5pxptP0XZ+f9XNawvbp9T8LCRoI0ktckBSNx+bgY6dKppqs1x4WgnnjtGKap5QK5xyo5HHoT71W0+e1bVvB2yJVY2pIYnHHz1l2ktmPBNr5kAP/ABN+CHAwdq+1EcPGfKmuq/8ATlfzKjCvFtKLXyXZf3jdnvJ7PWPF77YW2xIcbR3dAeSPeti0u7o6lpMZhiZX04SmYscj5eVzjmuY1N4BqXjMeWo/cRnG4dPMjq3ZSWp1zw0BAm7+yzk7hz8nH5Vy4jDqeEV19n/3Xj5lU41oVG+V79l/N/i/r8CkrtceGtKlSKNANYbdliT0j71Z1ETY8Z58r/Wxf+jKy7SS3XwnpQMaf8hlwDuHBxFzV3UHt9vjLECj95F3HA8wV6tWD+saL7X/ALsUznhQquLbg/8AwFfyv+9/W22po6Vv/wCEx0vO3H9hDP18lqyoBL/wjvhjPl4/tF+n++tX9Jkh/wCEy0nhQToIwdw/54txWTbyW/8Awj3hkiJR/wATJ+cj++tc0Iv20dOkP/ScQH1epytcj69F3j/e8v6635w/9jeNvuf8fseP+/rdasXIk/4SqTPl/wDIBbp/1xNUp5If7H8cAqv/AB+x5Gev75qnuJbdfFEh8tVH9gOeD1Hkmo5XzVHbpL/03QBYapy/A910XeX97+vysaYrHw74Rwtuzi+mxvfBHzjp607Vb1zoGs+VPY/u9QjUAwkgff8AvAg5P/16zbG7tG0HwV+73K1/cALg8fOOvpWffXdqNB8UHyzhdVjDDaeTmX2rWlBSxMm/+fj/APUmRoqFaMElTd/Rfyr+930/DzO1ugp8U37NPbbm0ZAytGRtHljJzjGOfyqpYSXBv/CUcc1s0bTyMxQE7sSk8HGB+nNU9SubX/hLNUBTcR4fVs4PTyVqhol5a/bPAoC4DTSbSVPy/vj+VedRpr6nf+55f8+KhvKFbS1P8PNf3vn/AFY3r641DTfDt/NDb6Yu69/v43gk/e77vatCae7j1/XvOijgjGm7kl3BgTtTORg9K4651m0t/DGsytp1rKI9XMZWSAkH7xyR1J960XubeTxZ4s+VQzaQSeMAjy4zj9a1q0lyTuv5vzoeZ01JVeVKMH9y/wDkv6/LVtLx21TwuJJrdlkQspAA3HOOMKP6Cqemzw2fg+QxxafcK+qqoEmdi5Trz/FWZp09t/aXgd1jCh0IU45zuP5VnW15af8ACuZG2fKNcQFdpznZ1xW8aMeaP+KH/p2sctq6lzKlr/hXZf3vl8jtb+zP2zX2zZJJJZD7kxL4ymSy4OB/9apbBXtpPDdsl5aHbY8qxkVpOTyo4H5jNc5fXdlBq/i9mgUstkh2kZ4/dgD37VZsNQtG1HwdGbSORpdOYrIVbcnLcA9B+NedOk1g4L+75f8AQMbU1L2zmqVn3su/lL57f5D2lS48N2lxbW1i5Opbmxv+9hefXdx9KteJYrm3bxYFMYjfY6/Pk8lQeMcdK463vLWPwlZFoiyjWmUgBhzsXngVueILi1/tPxyfJyU8oMeTn5l5H/1q9KtT5cVGz+0//T1LzOd0ak05Ok02v5V2a7/L5lzRpp28W+HldwUOjgke/lyVhxiX/hEtDz5f/ITb+YrQ0O6tz4z8NqFwW0UFSVPA8uTisWO6tP8AhENCfySAdUYZ2nrx7VVOyxEf+3P/AHYMfqtZxd4Pr0X93+95fh5a7VyH+y+PPuf8fK4/7/nrUd2H/wCE68M79mPLs87f97io7i4tzbfEH5PuXClsg/N+/P5/hTJp4D4/8LLGmxnitCpAPTcc/TPP51lH4av+Cf8A6YpDjhqunuPddF/M3/N8/wAfI57R/itrmlRRwnwhPeRQBFjdY2UlVGAdwXn61BffFrxXPcF08P3ERDEqr25bad5PoP4Tt/DPeuq8K/EnwhplvDHqOrpFJGkasDBuwVHI4XPBz+dYurePPCl3rD3EV2siGVm5jj5Bdj3TPTj/AIF6ivNy52x07UnHT4u/6Hm4+nCVH3sOp6vT9fmcN4n+Iuqa5pt/p95psUCzgK/yMHjwwY/xY5KjPH5V9f8Aw5/5JZ4U/wCwNZ/+iEr5W8XeNtB1Pw3qNlYs5nuIYkVTboFVkZSWV87uQCOnPWvqn4c/8ks8Kf8AYGs//RCV1O13aNtX8/M9eP8ADgrW0WnbyPnf9qj/AJHyy/7Btv8A+jbmvHLK4Ca3ZJPIFtxLF5gbG3bkZz7Yr2P9qj/kfLL/ALBtv/6Nua8Ouf8AX/8AAV/kKrXkaTt/TKTaR9XnQvBuoasLWz8P2klrNbFFu4LSMktz8wAGBxznv+h8+8A+GbO48beLNOjUTW1i2yE3Fv5mFw+DtPQ5wffFcF4R+LPiPwZp5tNNFpPHn5DdRs7Rj+6pDDjPaup8AeJNQvfEviHV2eSO7vh50gtdygsYpGwAG9RwOea+bwGW5jSdWnGpdNJRbd9U97dNAxmJhDDyqSvp2LHi7Qo7D406HpsPlwxzxglRb4UAs4AKA89MVs/Grw5Bpfg2zudNsre033ojzDAImOQdvIPsf06VyvinVbpPixpd9I1yJoVYhiZHYAFuR827HOcAjiuh+OepXM3h2xtpZcxNcO+wlznbtAIycfxH8+O9e6sFj1KnJzVor3vN2PI/tCk6+HhreabWr7de522qeD7GD4d3V39hgWUaYZXKw/MZBHncTnk571z/AMG9Atdc+H0d7f28VzP9qePzJIt7bRj5c54q1NquoS+AL7z2uFX+z/JQF32kGIHIGcY6jP1rmPhhqWq2/gOGGxW4MKyzSlondQGBQgEKec8/lisZYXNfq7XtFzSlp5K6RH16g6VV3doOzd/np/kJ4S0yC9+MniXSriETW1t53kwNDvWMg9QDwKX4jaLDpnxE8L6baW/kRXir5scEQUyfPjkdC3WqHg/VLqP4ya7PBJiW4SZjlnUsdhbA5z1A4P8AMVJ8Rr+6k8d+FbiZi1xGg4Ej5zkHG7OQc9wRXU8Lj/rHtFU921v+3r2v95ssdR+txpa3cb76dXt3NT4y6Jb6H4aiudNhFs5uEUmOFUIVlPBYdc7Tx7Gu8j8KWK+ERcrawrMbAS70h+ctsyDnPsa86+N19dzaJbQXHmrH9qZgskzsOAuCATjuwPf867JdUv5PDTeY9wI108D5ZXUfcTjGcd2/I1hHB5pGNOkqi5k9X5a/pc5ZZjh3hKdb3kntrd7217mH8FvD0ev+C7m71S3jupHvGVZZoxISmBkZzxzn8/esjSNEguP2g9Z0qSINbxxv+68nemAg6gnAHv8AT1p3wXvb6Lw3JbW3nCBr5mdonZcHavXB9qpaXf3K/tAalOguBLLuDBHZSP3ecHByRkDr7Vo8LmNSdWXtFyyVo+Ta3Z0zx1GnWqwd7xjzPXp5djW+IOjR6d4v8KafpqLapfStHKIoAofAjHI78kn8TVv4taAnh3wK1/psMNm4uo0DRWyoxBUZwQfUH8/as/4hX1y3jLwfcS/aPMjkZ13OxYnZGfl5yOfQ9c0nxj1W9uPCxgnlmML3EZVWLkEeWp5BYgHJP61KwOYxdOUqitG/N59TOhmdGo6EVf302tX5LXudvp/hawuvh7DqE9lC8s2lCR5DDl5G8vIJOecEVynwQ0O21fwXc3V7BHcSRXrxpJJFvKDapIBzkZyauWOuaong+2BW88hLOODKu+wo8QGQAQMD36Vi/B2+vrfwVcQ2nmbDfsT5cjKclV9D7Cs3hczdGp+8V2/d8lzJMc8bRhSqtt2i9de/YbFoVs37RkmkLbI1oIg32cxZT/Vg/d6fjVb4maKNH8SeGY44o4xdTtvCQBNwzHwyg/N1/pUf9oTxftELPHKyvJ5abpJHyVKgYznOfYnHaqfxC1i/uvEWhT3shM1tcsU3mQ7eI27nOM88fzroeExyrxm5rlSs/OXf7zopY2nKrCCvdxT3e3p3O2+KfhS00n4c3t9aWaW8kfkBZFhVGHQHkcjk8+9O8PeE7ef4W6bfPYwtM1i0zSNbKxYYGDnPXHf61m/FTVNQm8ASxzM/kSNb/enchv3e44BJzg4659etN8P65qUXwu02OKZjB5TxNskkwoyFAIB9CR6Z9q5Y5fmioxpuqudSu3/d7HJTzKhLDQqrms5W+J72/rQg+C/h6DW/C17c31t9pIuSis8IfHKE8n6/lmn+EvDdrcfHTxDpt1DHLBaxkJBJGGVenQdAB0/GqfwX1S7t/D+oW1s8hCyrII4ndWBLqCeDtIwPTPB55rP8O6lqUPxZ1+5tTKt28LgkO8rfeXvnJ6fhWk8Lj4yqVFNcso+6uzsdP12lKtWpXacUr6+XTsbfxT0K20jxt4OsrSBIIp58yJHHtViZFGcZrrviT4btdO+GetyWsK23lxhwIo9mTuHXnnr1rz/4gX97J4z8L/axJvtbxhEZXdiV3owJ3HPU9u2K6PxnqWp3Xw+1U34usTWz7zNI4AKyEAbScdMfpT+qZkpUYuotPi89v8zJ46ko0ZJv3mra73fXuXvBPg23v/AOnXctuksstm0nmPaqT0TjcTzj5vzPpXNfBLRLXWrXV2voEuBFOFj862EqoSexPfjp+NX/AARrGoD4e28Uf24xpHIgC3Em3jycYO7gYJIA/wBoVn/BK/ltrPXI4zcL++D74i20fMB2YDP4ZxnntULA5k4VYqoryfu+Sv8A5Cr5lQjSxE3zWg7PXztp2L+keHbWT4+63pU9rE9pDZiUQGEbBwpB25xn5v8AOaj+Lvh+30rUPC4tI47U3V2yyNHEU3YKjBGefSqOl6pfj4zeJLm2mkNybaSNXBYsAHAA65OMAUfGC8uJLnw2srys0NyWHmMzckqe5rohhMw9vFymuWKV13eq/MtY6l9ZpU1e8o6avtfU7X4meFNO0v4a6teWljDBJFErRNFDtKEsMnOf1qn8N/CtpqXw+0e5nsbWeadJDve1DE44G4k84OazPHms6nc+C9VW8F4I7iORD5jSFfkkyOM4GQM5xyMU74ea7qMHw5sUt7nEcPnqcGT5AIwyjg4zlu3r61xxwWaTwqj7Vc7d7+Xb79TCpmFCnhnP3mua3xPe3fsYfwj0ODWb/wARrewfaRDJsTdEH2/Njv04H+cVo6Jp8sH7QF7o0bOtmlu0qW0eRGMorAhAcDqKxfhBqN3BdeI/shkaV1MgRJHVidwGcg+565Pv1p2i6le/8L11O6SRjOYZlDKzkgAYHU5zgAYz9K6ZYXHqdSpGa5WrRXZ2/wCHO6pjqarVaet4xu9enl2Oo+MFpJo1voAsJpLcz6iqSeTuQMD68811XjnRRYeA9bu7fMc0Fkx3IpVlIHXO76VwPxUurmW38PfaWmJGpg4kkdsfMQOCeMiuj8W6pqFx4B19rtrjEkDxgM7Y24zkAn/Iohg80aox9orrWXnrf8jieaUlTpTUpWcrLV/zW17jPhFYyaz4Dsrm7mmuJmllTfINxXAOBuJ9SD9azPh7ZXWq+KPF8UtxcPFaXcyRq6+YkfLBSATxjHAFM+EOo3sXgSOKBZzEs0wO24ZQf3eRgA4Xnj681Q8Banc2vi7xgkH2jLz3EmYGbduGfRhzzxkHn0qJ4LM5RqONRLmty+W7/IqWa0Yzrp83uWv73y07F2W3nP7RkejPcSy2c0LF7c5EZBjLY25we1Q/HjQrbQ/DukzWcCW8jXTJuij2fKFyB16Csi91G9T4+TXUBn+0pFIIySzN/qzjHOfoM0nxhv7650Gzt79bj93dCRHlZyH3Ic43E4wR0HTNUsNj1VhzTXLGPvebs0dCxlOdSjZu8o6a9Lde53nhTwz4Yg8G2ElxpUdzK5inllurNS2SisUUkfMpyTnoK8p+M02kwePjF4YS3trRLSMPDbRhFSTLZBA79KW5+NHimLQRoES2KW0ESwxSiE+agVQoIO7GcD0rzqSR5ZGklYu7ElmY5JPrXj4HA4uljJ4ivN7uyu2rN7nsVaqa5Ykl5/x9v+H8q++fhx/ySzwp/wBgWz/9EJXwNef8fb/h/Kvvn4cf8ks8Kf8AYFs//RCV9BP4mc8t2eC/tNaPfaj44s5LO3kkUadApKxswyJLj0B9RXjI0DXgABbTYAwP3Df/ABNfoJRSUmtgu1sfn3/YOv8A/PtN/wB+G/8Aia7H4bxXOi61dPrYngt54Cpc2sjgPkbTgIeQN2DjivtOilKUpRcbsTtJWkrrzPizxCl1dfEiHVLD7S9lBIvlyi3lGF4zwUHfdnjqTWl8SLq+8R6fawWk0l2FldmSK0mRVX5NmdyDJ+/+dfYNFNTmre89Px6akunTbT5VdbaLT07HyZJeyH4dyWRu7iS9ksUgFsbGbcjCQZO/Z3TIPPTFV/hrMdB0W6g1Z5reRZVeCJ7SZ0bqGziM8Y/XFfXdFTKVSSa53r1v+AlTpJcvIrPXbqfFHhaC+sPGlzqN4kiJMJMTT2ksg3FWwSAhPUjtUnjGK+1Lxfa3dg9zdW9sq7JRayIFOBuwpQd89q+06Kv2k+bmv/V7lcsObm5Vf0PkX4iTPrnhe0srG6utRuEui7ZspYwsePl6xgZ5xx6VZvdQJ+GsGnWU04vzaJHPG1pNv3hxwrbMAbc5OfavrGis05q3vvR3339SVSpKKioKy8kfHHw7W50a1vLfULmbT45ZFdSbGWXP3skAIQD93PqPpVLS4byL4k3GqzCeO1mmkPnSWsrgod2MrsJx04xX2pRVuc227va3oPkp3cuVXe+m58cePDLqfivTLnRDdz2tnGv7w20qkSZO5sFByfl5x2HpV34mX954i00WmmzSXEHn7lhjtJ02qOm7cgB/Cvrqipi5xt7z0v13v3F7Olp7i020Wnp2Pkm1vJ4vh8LM3k63i2skAtvsUxY5b5Tv2cEDgc4warfDIroNhqEOsGe1aR43hBtJnDEb8jhDgZK59RX1/RROVSUWud69b6r0H7OlquRWe+m58V6FDcw/EltY1OCR7ZmmPmTWsrrk7tmV2E4yV7cUvjiPUNV16wuLB57lbaJSZI7WWNVkySxUFAf7vNfadFXzz5ua4+WF0+VXXkfJPj27l1fwqLHT726vXacN5Rs5Y9kew/KcxgHDYxUnhu6i034ftZXBmW/aGeMxNZynblTsKkR4zkjknoK+sqKzfO48rk977/h6CjTpRjyqCt2svv8AU+PPhrLd+HrO6t72aWwEsqNuNnNIduW3bcIcH7vX0FZujQ38PxDuNTuJbq3gnMim6+yyMcFWCkrsOf4eMV9q0Vp7Sd3aT/y9A9nTu3yq73039T4u+IMNxrPiKzuNKN1cwwQKDILaVAsmWLbQUGByO3pW18RbyTWtIS10WSSWIuCYls5kYjYc7iUAI3HgZPavreipUppxfM9PPf1Dkp6e6tNtFp6dj5G0adrTwDHaGeW2voYZQIRYSlpGO7b8/lkDGUIOe1U/hm1zoF1erqk91Y20zBjstZWL89sIcH619i0Uc0+WUeZ6/h6CdOk7pwTT3039T4xU3v8AwtebW1+1Cye7YiVreXBiy20FdmcY28YpPiEl5r2oabLZh7gwR/vWhtJo1DeY5GAyDnaV5x2r7Popqc00+Z6f1qVywUlJRV1tofJHi66l1PwYbOzu7i6uJhDvh+xSoUwhDgtsG4bsHknmneELpdK8ES2120sWoLvWKNrOY/KUGDkRnkN05GOa+taKlupKPK5Pe976+noJU6SjyqCt2srHxx8OnuvD76u10Xs5bpkMUslnM/G4lsbUODiq1ol1/wALXbWbo3ZsmuXJuTbSE7MsFO3Z6Y4x+FfaFFU5z1956/h6ByU3Jy5Vd6PTdHxf4+hu9Y1HTvsM1zfRwQp5ji1ljXzAxyQpQc42mt3XL03vw2TTIpZ5r5rW3WRGtJvM80Y3gvsA28epzX1nRScptRTk9PPf1D2dNJJQWm2i09Ox8heASdJ0P7Nqk72hSYv5Z0+WRnGF4yEIwfmHrVPwC19pWuX9zqlzdWa3I3NKbWVzIxIz0QkHqc+tfZNFJyn73vPX8PQXs6et4p330WvqfGF2Lub4tJrX+mPp63S4nNvLu8kMM8bM/dzxil+JC33iE2q2Za6SMsdsNpMirnGMhkGTx1r7Ooqued17z2t6j5Kd01FabaLT0Pz8/sDxAettOfrbv/8AE0f2Br//AD6zf+A7f/E1+gdFPnl3L5n3Pz2fwvrUjl3s5yx6nyX/APia+6Ph9E9v8M/DEMqlXj0i0VgRggiFQa6Gip3Ef//Z)**

**Authors:**

Niv Bar-Tov, 206133829

Dor Lerman, 209400761

**Supervised by:** Goel Samuel

This project was done in collaboration with Apple Israel.

**Table of Contents**

**List of Figures**

**List of Tables**

1. **Abstract**

The Local Sequence Alignment Problem, where two sequences are compared to identify similarities between them, is a fundamental tool in bioinformatics, genomics, proteomics, and other related fields. The Smith-Waterman algorithm is a dynamic programming algorithm which solves this problem. However, its computational complexity poses a significant challenge when dealing with large genomic databases. This project focuses on the design, implementation, and verification stages of planning a dedicated hardware accelerator for the Smith-Waterman algorithm, aimed at significantly improving its execution speed from a quadratic complexity to linear complexity.

The proposed hardware accelerator leverages parallel processing and custom hardware architecture to accelerate sequence alignment tasks. We discuss the architectural plan, hardware design, logical simulations, and alternative solutions. Our approach emphasizes optimizing time efficiency to make the hardware accelerator suitable for resource-constrained environments.

Our project's target is to provide a high-performance solution for local sequence alignment, addressing the growing demand for rapid and accurate sequences analysis.

1. **Introduction**

Numerous fields presently necessitate sequence database searches that rely on pairwise alignment. This involves comparing a given query sequence with a database of sequences to pinpoint the sequence with the highest degree of similarity. In the realm of bioinformatics, a sequence alignment is a way of arranging sequences of DNA, RNA, or proteins to identify regions of similarity that may indicate on a relationship between the sequences. Such similarities can yield valuable insights into the function of the query protein or a gene. Furthermore, this method is also beneficial for alignment of non-biological sequences.

In the case of DNA sequences, the residues are nucleotides that each one of them can be one of the four nucleobases: adenine, guanine, cytosine, and thymine. These nucleotides will be called throughout this project as A, G, C, and T, accordingly.

* 1. **Global and Local Alignment**

Computational approaches to sequence alignment generally fall into two categories: global alignment and local alignment. Performing a global alignment constitutes a type of global optimization that ensures the alignment extends across the entire length of the sequences. On the other hand, local alignment pinpoint regions of similarity within sequences that are often widely divergent overall. While local alignments are frequently favored, they can pose greater computational complexity due to the added task of identifying these similarity regions. Table 1 summarizes the differences between these two approaches.

|  |  |
| --- | --- |
| **Global Alignment** | **Local Alignment** |
| Align the entire sequence | Align regions having highest similarities |
| Suitable for closely related sequences | Suitable for more divergent sequences |
| Main well-known algorithm: Needleman-Wunsch Algorithm | Main well-known algorithm: Smith-Waterman Algorithm |

Table 1: Differences Between Global and Local Alignment

* 1. **Motivation and Alternative Solutions**

While short or similar sequences can be manually aligned, the alignment of longer, highly diverse, or numerous sequences often exceeds human capabilities. To tackle such challenges, computational algorithms come into play. These include slow but formally correct methods like [dynamic programming](https://en.wikipedia.org/wiki/Dynamic_programming). Aligning just a few hundred DNA or protein sequences can demand several CPU hours on high-performance computers. Alternatively, there exist efficient heuristic algorithms and probabilistic methods tailored for large-scale database searches, although they do not guarantee to find the optimal matches.[1][2] Hence, there is a high motivation to implement hardware accelerators to keep up with the increasing amount of data.

|  |  |  |
| --- | --- | --- |
| **Solution** | **Advantages** | **Disadvantages** |
| Alignment by hand | * No need of computing resources * Efficient for very short sequences | * Slow method * High potential of mistakes * Impossible to do on long sequences |
| Software implementation of dynamic programming algorithms | * Accuracy: guarantee to find the optimal alignment | * Slow method for very long or highly variable sequences * Requires expensive computing resources |
| Heuristic algorithms and probability methods | * Fast and efficient method * Consumes few computing resources | * No guarantee to find the optimal alignment |
| Hardware accelerator implementation | * Fast and efficient method * Consumes few computing resources * Accuracy: guarantee to find the optimal alignment | * Need of dedicated hardware to perform the alignment |

Table 2: Comparison of Alternative Solutions

* 1. **The Chosen Solution**

We decided to implement in this project a hardware accelerator that will solve the local sequence alignment problem using Smith-Waterman algorithm, which is one of the well-known algorithms for solving this problem by [dynamic programming](https://en.wikipedia.org/wiki/Dynamic_programming). The accelerator will be separate from other expensive computing resources and will solve the problem more efficiently because the computation will be done parallelly and directly on the hardware, instead of complex programming implementations. The hardware implementation is done in linear time complexity , while the best software accurate implementation is done in quadratic time complexity . The space complexity is quadratic , like the software implementations.

The goal of this project is to perform the architectural plan, frontend design and verification stages of the hardware accelerator that implements Smith-Waterman algorithm for local sequence alignment. The sequel project will cover the backend implementation stages of the accelerator.

1. **Smith-Waterman Algorithm**

Smith-Waterman algorithm is a dynamic programming algorithm used for local sequence alignment.[3] It compares segments of all possible lengths and optimizes the similarity measure, using a substitution matrix. Its time and space complexity are quadratic.

* 1. **Scoring System**

The algorithm assigns each cell in the substitution matrix (pair of residues) a score based on the scoring system. The following definitions are required to define a scoring system:

* The **match/mismatch score** is the similarity score of the two residues related to the cell . In case the two residues are identical, the match/mismatch score is . In any other case, .
* **Gap penalty** is the penalty of a gap in the alignment. The value of the gap penalty is .

It is possible to define another scoring system with different values for the match/mismatch score and the gap penalty. We defined the scoring system described below for the implementation of the accelerator.

|  |  |
| --- | --- |
| **Score** | **Value** |
| Match | 1 |
| Mismatch | 1 |
| Gap Penalty | -2 |

Table 3: Chosen Scoring System

* 1. **Main Algorithm Stages**

Let be the query sequence to be aligned with  
, the database sequence. The Smith-Waterman Algorithm consists of 3 main stages:

1. Initialization: Construct a substitution matrix and initialize its first row and column with zeros. The dimensions of the matrix are  
   .
2. Scoring: Fill the substitution matrix and calculate the score for each cell based on the scoring system using the following equation:
3. Traceback: Determine the optimal alignment by trace back from the highest-scoring cell in the matrix to a cell with a score of zero. We will generate the best local alignment in reverse direction by trace back recursively based on the source of each cell. There are 3 different options for each traceback step:

* Source is the diagonal adjacent cell – both aligned sequences contain the residue related to the cell.
* Source is the left adjacent cell – the aligned query sequence contains the query residue related to the cell and the database sequence contains a gap (–).
* Source is the top adjacent cell – the aligned database sequence contains the database residue related to the cell and the query sequence contains a gap (–).

A cell can receive score from more than one adjacent cell, each will form a different path if this cell is traced back. In case of more than one cell with the highest score, the optimal alignment is not unique (all these alignments have the same similarity measure).

Using this algorithm stages ensures the optimal local alignment of the two sequences.

* 1. **Example**

Let TACGCTTG be the query sequence and CTACCTAG be the database sequence. The initialized substitution matrix appears in figure 1.

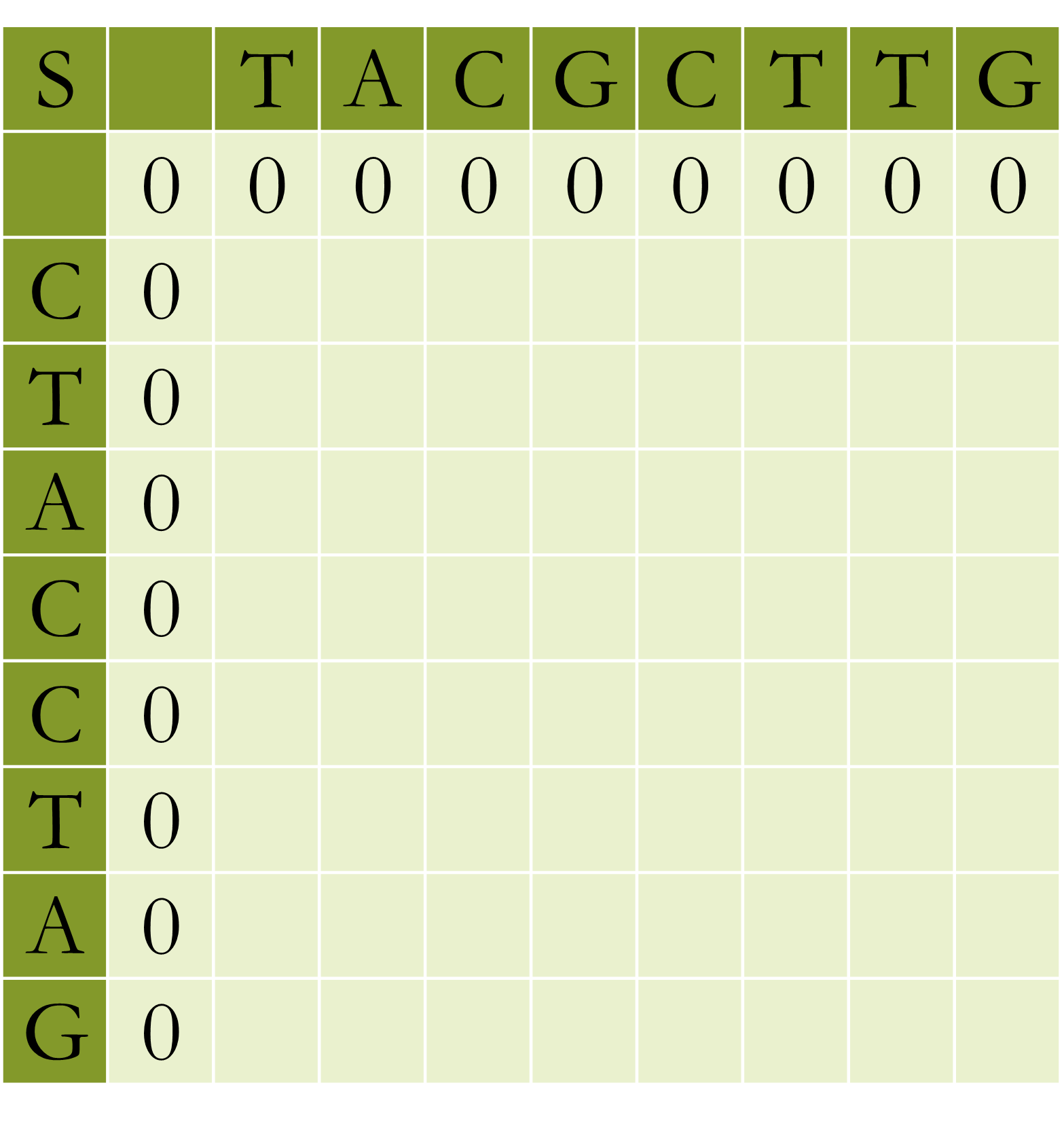


Figure 1: Initialization of the Substitution Matrix

According to the scoring system, cell has a mismatch and . Thus, the cell's score is 0. Figures 2 and 3 show the calculation process of the substitution matrix.

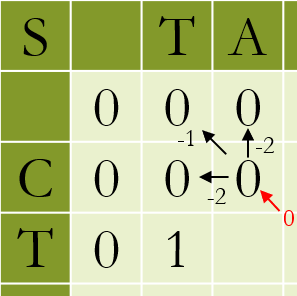
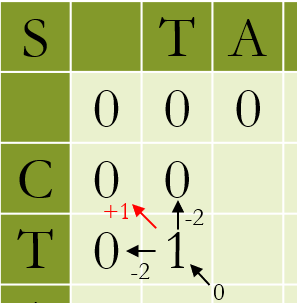
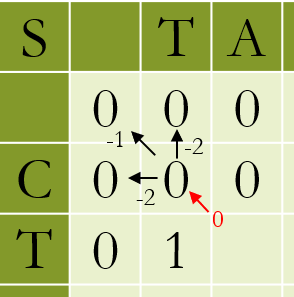


Figure 2: Calculation of 3 Cells in the Substitution Matrix

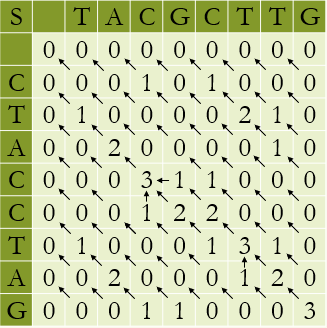


Figure 3: Calculation of the Substitution Matrix

After filling the entire matrix, apply the traceback process and generate the optimal alignment. There are 3 different optimal routes, shown in figure 4.

(a) (b) (c)

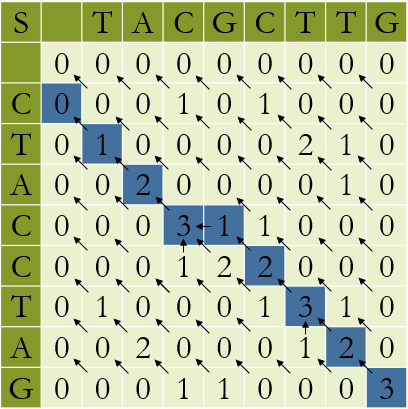
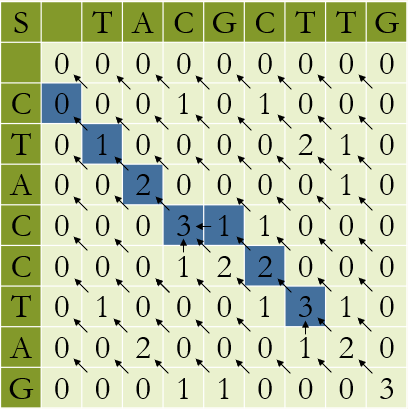
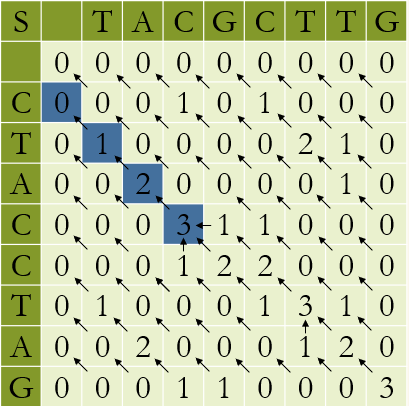


Figure 4: Traceback Possible Routes

Each one of these routes generates a different optimal alignment:

TAC TACGCT TACGCTTG

TAC TAC–CT TAC–CTAG

(a) (b) (c)

1. **Goals, Requirements and Specifications**

The project has several requirements and specifications we defined at the beginning of the work:

* The design will be implemented based on Smith-Waterman Algorithm.
* Each query and database sequence will contain 32 residues exactly (no support for alignment of sequences with different lengths).
* The system will not support alignment of more than two sequences in parallel.
* The scoring system will be predetermined and unchangeable (it will be part of the design).
* The system will output one possible pair of aligned sequences in case of more than one optimal route. It will output the similarity score as well.
* The accelerator will be designed in a multi-cycle architecture: each pair of sequences will be aligned in more than one cycle, without pipeline.
* The accelerator will be implemented using SystemVerilog hardware description language.
* The accelerator will be fabricated in TSMC with a technology process of 65 nm.
* The chip uses a single clock. The target clock frequency is up to 300 MHz.
* Aim to achieve maximum area efficiency, with area target of 1.0x1.0 sq. mm.
* The maximum number of I/O pins is limited to 40.

There are many ways to optimize the chip or make it more general and configurable according to the Smith-Waterman algorithm. For example, it could be possible to design the accelerator in a pipelined architecture to increase the system's throughput. It is also possible to support variable scoring system or alignment of sequences with unequal lengths. However, we had to simplify the project and to meet the time frame of the project work. Timing, power, and area constraints were also considered.

1. **Top Level Architecture**
   1. **General Terms**

Throughout this project, we will use the following terms and definitions:

|  |  |
| --- | --- |
| **Term** | **Description** |
| **Processing Element (PE)** | The basic computation unit of the system, performs computation of a single cell in the substitution matrix. |
| **Processing Unit (PU)** | Consists of 4 PEs, which are arranged in a 2x2 matrix. |
| **Sequence** | An input sequence consists of 32 letters. A letter can be A,T,C or G. |
| **Aligned Sequence** | An output aligned sequence consists of 32 letters or a gap. |
| **Query Sequence** | The top Sequence in the matrix. |
| **Database Sequence** | The left Sequence in the matrix. |
| **Data Packet** | Information regarding a single cell in the substitution matrix. The information includes the cell's adjacent source and a zero-score bit. |

Table 4: General Terms and Definitions

* 1. **Top Level Design Interface**

The accelerator gets two sequences as an input, each one of them consists of 32 letters, 4 letters from each sequence in a cycle. The accelerator will compute the best possible alignment of the sequences and will output the aligned sequences, one letter from each sequence in a cycle (in a reverse order), as well as their similarity score.

תמונה שמכילה טקסט, צילום מסך, כרטיס ביקור, גופן

התיאור נוצר באופן אוטומטי

Figure 5: I/O Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **Signal Name** | **In/Out** | **Size (bits)** | **Description** |
| clk | input | 1 | System's clock. |
| rst\_n | 1 | Reset signal, active on low phase. |
| query\_sequence\_in | 4x2 | Sub-sequence of 4 letters of the query sequence. |
| database\_sequence\_in | 4x2 | Sub-sequence of 4 letters of the database sequence. |
| start | 1 | Single pulse signal which indicates the start of sequences loading. It must be active one cycle before loading the sequences. |
| ready | output | 1 | System is ready to get a new pair of sequences as an input. |
| query\_seq\_out | 3 | One letter of the aligned query sequence. |
| database\_seq\_out | 3 | One letter of the aligned database sequence. |
| score | 7 | The maximum score of the optimal sequence alignment. |
| output\_valid | 1 | Indicates the output is valid and ready for read. |

Table 5: Top Level I/O Signals

* 1. **Block Diagram**

The top-level unit consists of the following sub-units:

1. **Sequences Buffer** – stores the inserted sequences.
2. **Matrix Calculation** – responsible for filling the substitution matrix by calculating the score of each cell.
3. **Matrix Memory** – stores the information regarding each cell in the substitution matrix. The Matrix Memory is based on registers.
4. **Max Registers** – responsible for computing the maximum score and index (position) in the substitution matrix. It is updated after every cycle of computation, if necessary.
5. **Traceback** – responsible for recursively trace back from the maximum cell to a cell with a score of zero. It will generate the optimal local alignment in a reverse order by trace back recursively according to the source of each cell.
6. **Controller** – FSM that provides control signals to all the sub-units.

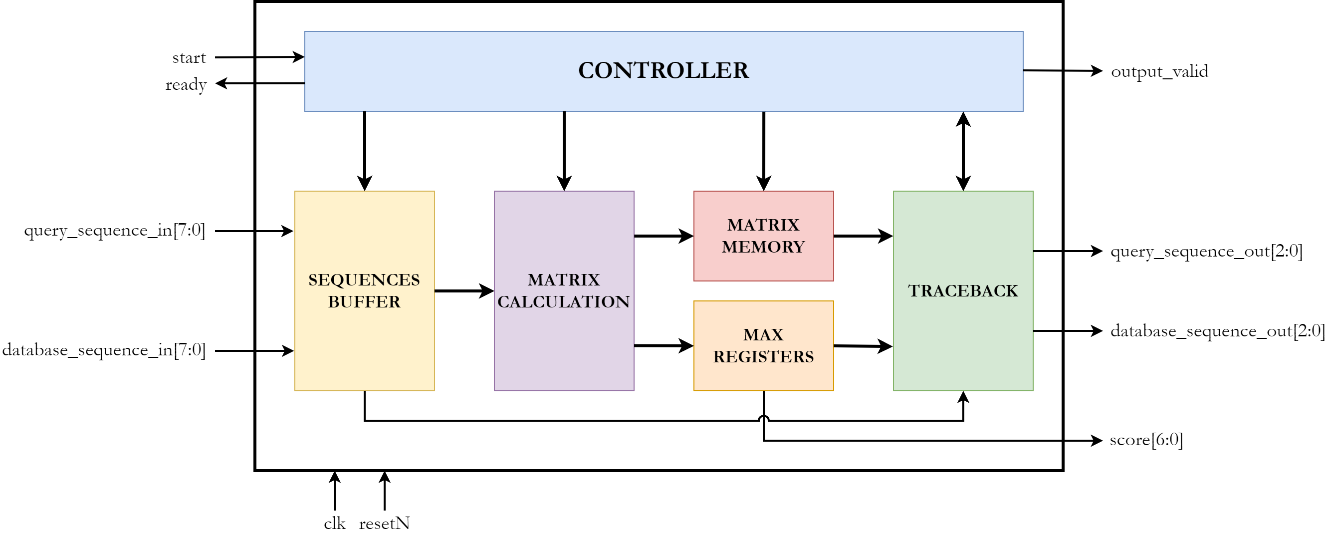


Figure 6: Top Level Architecture Scheme

* 1. **High Level Architecture Stages**

The Alignment process of the accelerator consists of 4 main stages:

1. **Idle** –Signifies that the chip is awaiting a start signal, indicating its readiness to perform a sequence alignment task. To proceed to the next stage, a "handshake" of 'start' and 'ready' signals is required.
2. **Load Sequences** – Load the two input sequences into the Sequences Buffer Unit in parallel - 4 residues from each sequence (query and database) on each cycle.This stage takes 8 cycles to store the 32 letters sequences.
3. **Score Matrix Computation** – The substitution matrix is filled using the Matrix Calculation Unit, with updates happening concurrently in the Matrix Memory and Max Registers Units. This process begins one cycle after the start of the previous stage and continues for a total of 32 cycles.
4. **Traceback** – Start at the highest-scoring cell in the matrix and follow the path of the highest-scoring cells until a cell with a zero score is reached. This is done by the Traceback Unit. Traceback stage takes between 1 to 31 cycles depending on the number of steps taken on the route to a zero-score cell.

Figure 7 displays a timeline with stages and their respective active units. The Controller Unit controls and schedules the computation stages.

תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי

Figure 7: High-Level Architecture Stages

* 1. **Letters Representation**

The input sequences are represented differently compared to the aligned output sequences, because of the need to represent a gap in the aligned sequences.

|  |  |
| --- | --- |
| **Letter** | **Value** |
| A | 00 |
| G | 01 |
| T | 10 |
| C | 11 |

Table 6: Input Sequences Represantation

|  |  |
| --- | --- |
| **Letter** | **Value** |
| A | 000 |
| G | 001 |
| T | 010 |
| C | 011 |
| – | 100 |
| Don’t care | 101 |
| Don’t care | 110 |
| Start/End Signal | 111 |

Table 7: Output Sequences Represantation

* 1. **Data Packet**

Each cell in the substitution matrix contains the following necessary information:

1. **Zero Score Bit** – Indicates if the cell's score equals zero.
2. **Source Bits** – 2 bits indicate the adjacent cell from where the maximum score has been determined (left, top or diagonally located).

Data packets are kept in the Matrix Memory Unit, and they are used by the Traceback Unit. To reduce area, only the source and the zero-score bit of each cell are stored, instead of keeping the whole score.

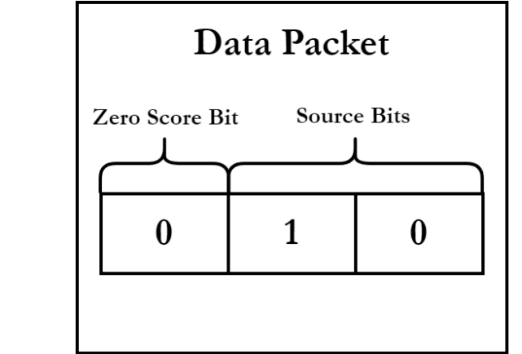
****

Figure 8: Data Packet

1. **Sequences Buffer**

The Sequences Buffer Unit stores the query and database sequences. It receives them in 8 consecutive cycles, while it stores 4 residues from each sequence on each cycle. The unit is controlled by the 'count' signal, which enables writing for the appropriate registers. Figure 9 shows the architectural scheme of the Sequences Buffer Unit.

תמונה שמכילה צילום מסך, מלבן

התיאור נוצר באופן אוטומטי

Figure 9: Sequences Buffer Scheme

* 1. **Interface and Signals**

The Sequences Buffer Unit gets the sequences letters as design inputs and control signals from the controller. The outputs of the unit are the stored sequences that are used by Matrix Calculation Unit and Traceback Unit.

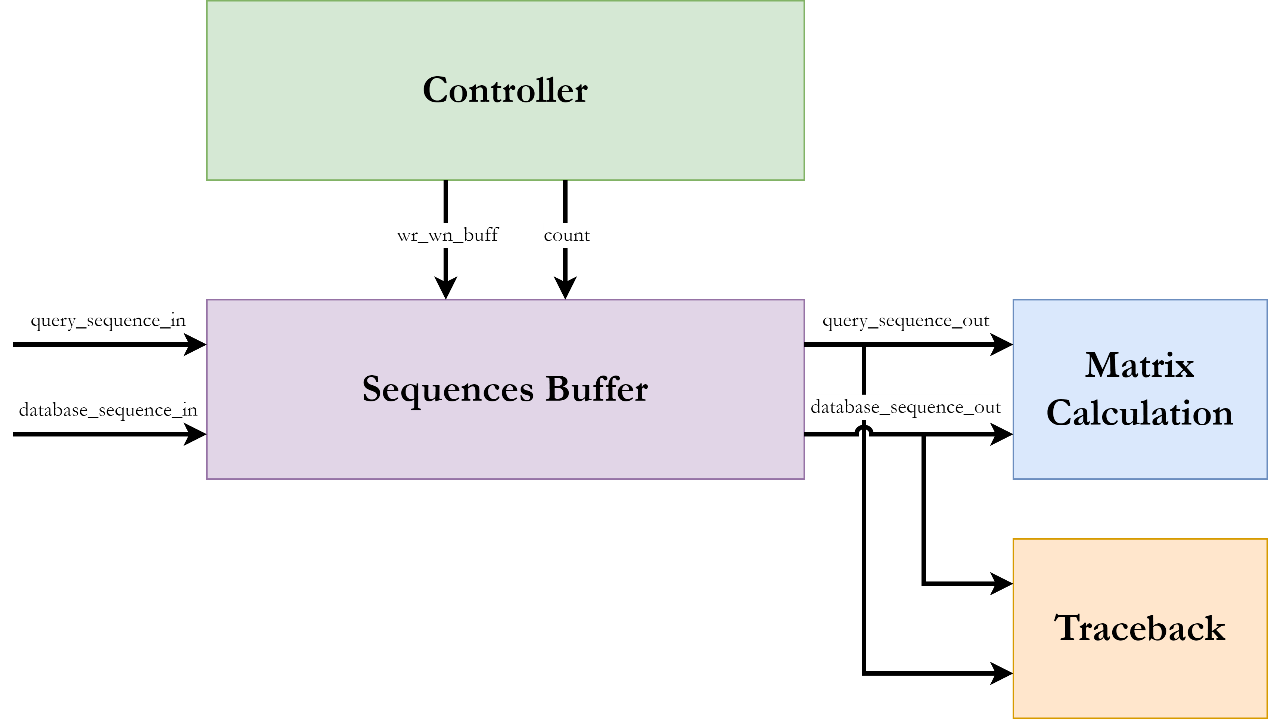


Figure 10: Sequences Buffer Interface

|  |  |  |  |
| --- | --- | --- | --- |
| **Signal Name** | **In/Out** | **Size (bits)** | **Description** |
| clk | input | 1 |  |
| rst\_n | 1 |  |
| wr\_en\_buff | 1 | Enables loading of sequences. |
| query\_sequence\_in | 8 | 4 residues of the query sequence. |
| database\_sequence\_in | 8 | 4 residues of the database sequence. |
| count | 3 | Controller's signal that counts from 0 to 7. |
| query\_sequence\_out | output | 2x32 | 32 letters of the query sequence. |
| database\_sequence\_out | 2x32 | 32 letters of the database sequence. |

Table 8: Sequences Buffer I/O Signals

1. **Matrix Calculation**
   1. **Processing Element**

**תמונה שמכילה צילום מסך, עיצוב

התיאור נוצר באופן אוטומטי**The Processing Element serves as the fundamental computational unit in the design. A single Processing Element (PE) represents a cell in the substitution matrix. The PE does the basic operation of computing the score of the current cell: it takes the top, left and diagonal scores from the adjacent cells and performs the calculation based on the scoring system described in section 3.1.

Figure 11: Processing Element Scheme

* + 1. **Processing Element Interface**

|  |  |  |  |
| --- | --- | --- | --- |
| **Signal Name** | **In/Out** | **Size (bits)** | **Description** |
| top\_score | input | 7 | Score value of the top cell. |
| left\_score | 7 | Score value of the left cell. |
| diagonal\_score | 7 | Score value of the diagonal cell. |
| query\_letter | 2 | The related letter from the query sequence that the PE uses to determine if it is a match or a mismatch is exist. |
| database\_letter | 2 | The related letter from the database sequence that the PE uses to determine if it is a match or a mismatch is exist. |
| max\_score | output | 7 | The maximum score out of all 3 directions: top, left and diagonal. |
| source | 2 | Direction from where the cell's score was updated. |

Table 9: Processing Element I/O Signals

* + 1. **Max Unit**

The Max Unit is a basic unit that outputs the maximum score among three values (or zero) and provides the source of this maximum score.

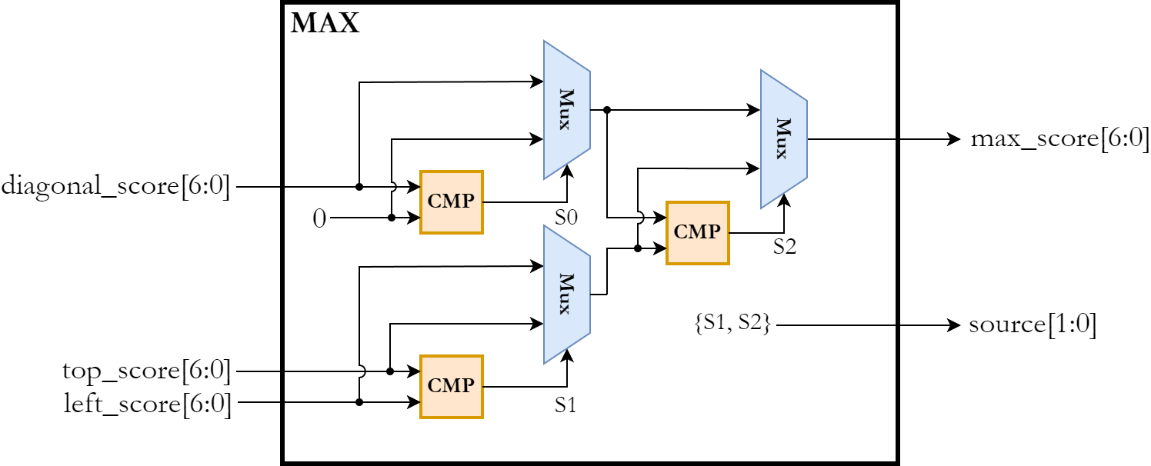
****

Figure 12: Max Unit Scheme

The possible values of the source are:

* 2'b00 – source is the diagonal adjacent cell
* 2'b01 – source is the left adjacent cell
* 2'b11 – source is the top adjacent cell
  1. **Processing Unit**

The processing Unit (PU) is a structure of 4 PEs arranged in a 2x2 matrix.

תמונה שמכילה טקסט, צילום מסך, גופן, מספר

התיאור נוצר באופן אוטומטי

Figure 13: Processing Unit Structure

The Processing Unit calculates the data packets of the 4 cells in a single cycle. The PU can send data to other PUs by providing the scores of the internal PEs calculated in the current cycle. Given its composition of four PEs, the PU derives its value from the top, left and diagonal directions. Hence, for each PE, an indicator for a zero calculated score is sent to the Matrix Memory Unit, together with the source of each cell. Packing the zero-score indicator with the cell's source forms a data packet as described in chapter 5.6.

**תמונה שמכילה צילום מסך, פתקייה, מלבן, ריבוע

התיאור נוצר באופן אוטומטי**

Figure 14: Processing Unit Scheme

* + 1. **Processing Unit Interface**

|  |  |  |  |
| --- | --- | --- | --- |
| **Signal Name** | **In/Out** | **Size (bits)** | **Description** |
| top\_score | input | 2x7 | Score values coming from the top cells (2 scores in total). |
| left\_score | 2x7 | Score values coming from the left cells (2 scores in total). |
| diagonal\_score | 7 | Score value coming from the diagonal cell (a single score only). |
| query\_letters | 2x2 | 2 letters of the query sequence related to the current PU cells. Used to determine if a match or a mismatch is existing. |
| database\_letters | 2x2 | 2 letters of the database sequence related to the current PU cells. Used to determine if a match or a mismatch is existing. |
| scores | output | 2x2x7 | The scores of the 4 cells computed by the PEs. |
| sources | 2x2x2 | Direction from where the score of each cell was determined. |
| zero\_score\_bits | 2x2 | Indication if the score of a PE is equals zero. |

Table 10: Processing Unit I/O Signals

* 1. **Matrix Calculation**

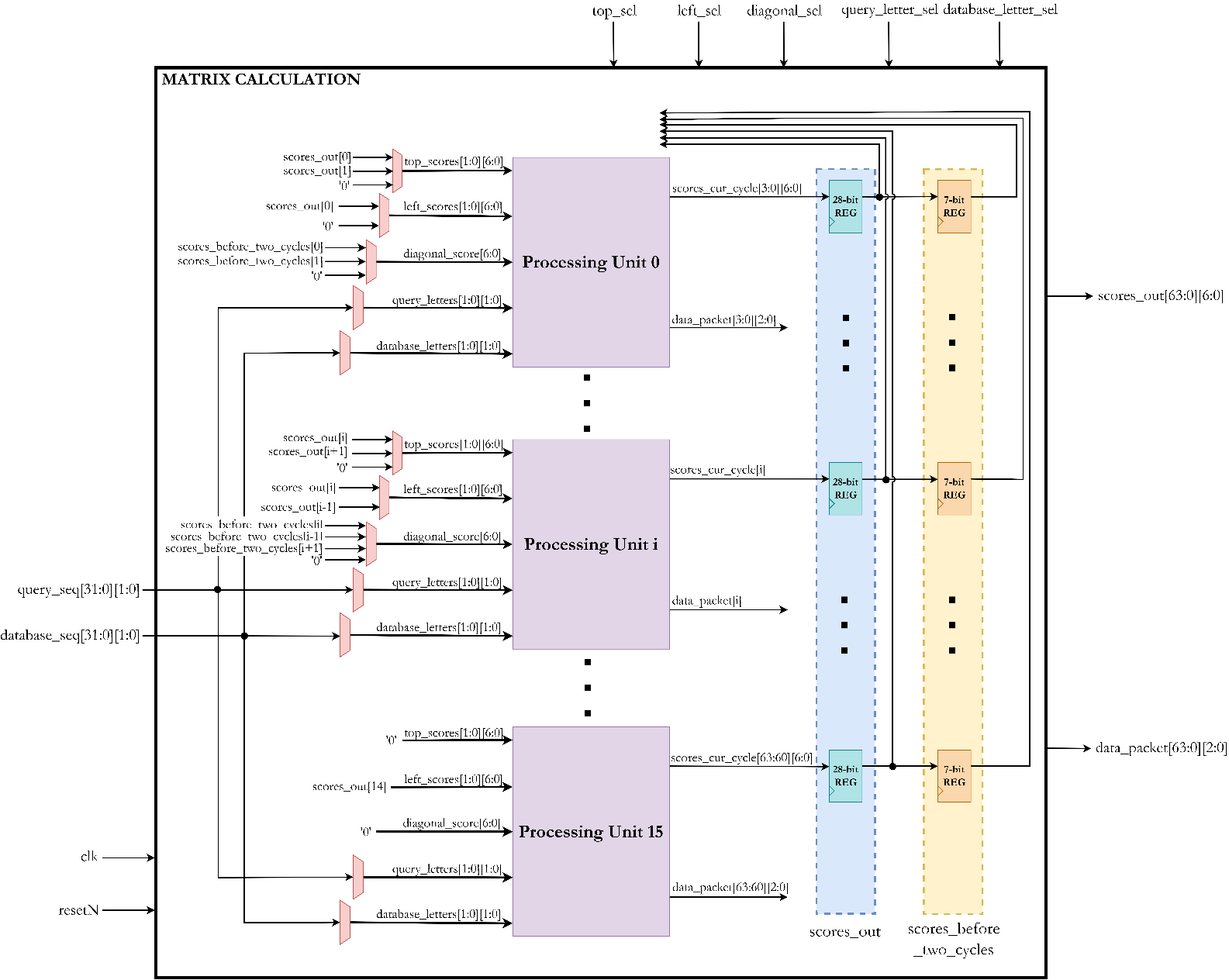
The Matrix Calculation Unit is responsible for performing the substitution matrix score computation.It consists of 16 identical PUs that perform the same task: compute scores and assemble data packets. The amount of 16 PUs serves as a strict lower limit for the PUs to work in parallel along the main diagonal. Since this Unit consists of a PUs array, the computation is done in the granularity of PUs. The 16 PUs are connected to each other and can work together in parallel to provide the required data on each cycle.

Figure 15: Matrix Calculation Scheme

* + 1. **Matrix Calculation Interface**

|  |  |  |  |
| --- | --- | --- | --- |
| **Signal Name** | **In/Out** | **Size (bits)** | **Description** |
| clk | input | 1 | Design's clock. |
| rst\_n | 1 | Reset, active on low phase. |
| query\_seq | 32x2 | Query sequence, consists of 32 letters. |
| database\_seq | 32x2 | Database sequence, consists of 32 letters. |
| top\_sel | 16x2 | 16 select signals for the 16 PUs, determines the inserted values from the top direction. |
| left\_sel | 16x2 | 16 select signals for the 16 PUs, determines the inserted values from the left direction. |
| diagonal\_sel | 16x2 | 16 select signals for the 16 PUs, determines the inserted value from the diagonal direction. |
| query\_letter\_sel | 16x2x5 | 16 select signals for the 16 PUs, determines the 2 related query letters. |
| database\_letter\_sel | 16x2x5 | 16 select signals for the 16 PUs, determines the 2 related database letters. |
| wr\_en\_pu | 16 | Enables the registers of the PUs output. |
| scores\_out | output | 16x2x2x7 | The scores output from the 16 PUs. |
| data\_packet | 16x2x2x3 | The data packets output from the 16 PUs. |

Table 11: Matrix Calculation I/O Signals

* + 1. **Computation Flow**

The computation starts from the top left corner in the matrix and ends in the bottom right corner. Figure 16 describes the direction flow of the computation (each color refers to a different calculation cycle). It is done in a diagonal manner: on the first cycle, we calculate the top left diagonal and use the first PU only. Then, on the second cycle, we calculate the next diagonal while using the first two PUs. The calculation continues until we get to the 16th cycle .On this cycle the unit computes the main diagonal and uses all the 16 PUs. On the 17th cycle the unit computes the next diagonal and uses 15 PUs, and so on, until we get to the 31st cycle, in which the unit computes the bottom right diagonal using a single PU.

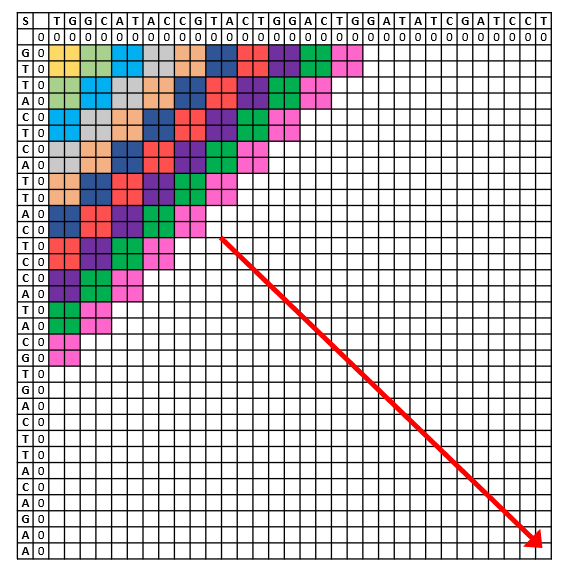


Figure 16: Matrix Computation Flow

* + 1. **Interaction Between PUs**

Assume PUi from the PUs array performing a calculation. For that, it needs 5 scores of the adjacent cells, as described in figure 17.

תמונה שמכילה צילום מסך, ריבוע, מלבן, קו

התיאור נוצר באופן אוטומטי

Figure 17: PU's Needed Scores

Figure 18 illustrates the interactions of a PU with index of i with other PUs in the PUs array, and the data selection performed by the multiplexers.

This illustration presents Units number i , when . Units number 0 and 15 are considered to be a private case of the general case presented in this figure.

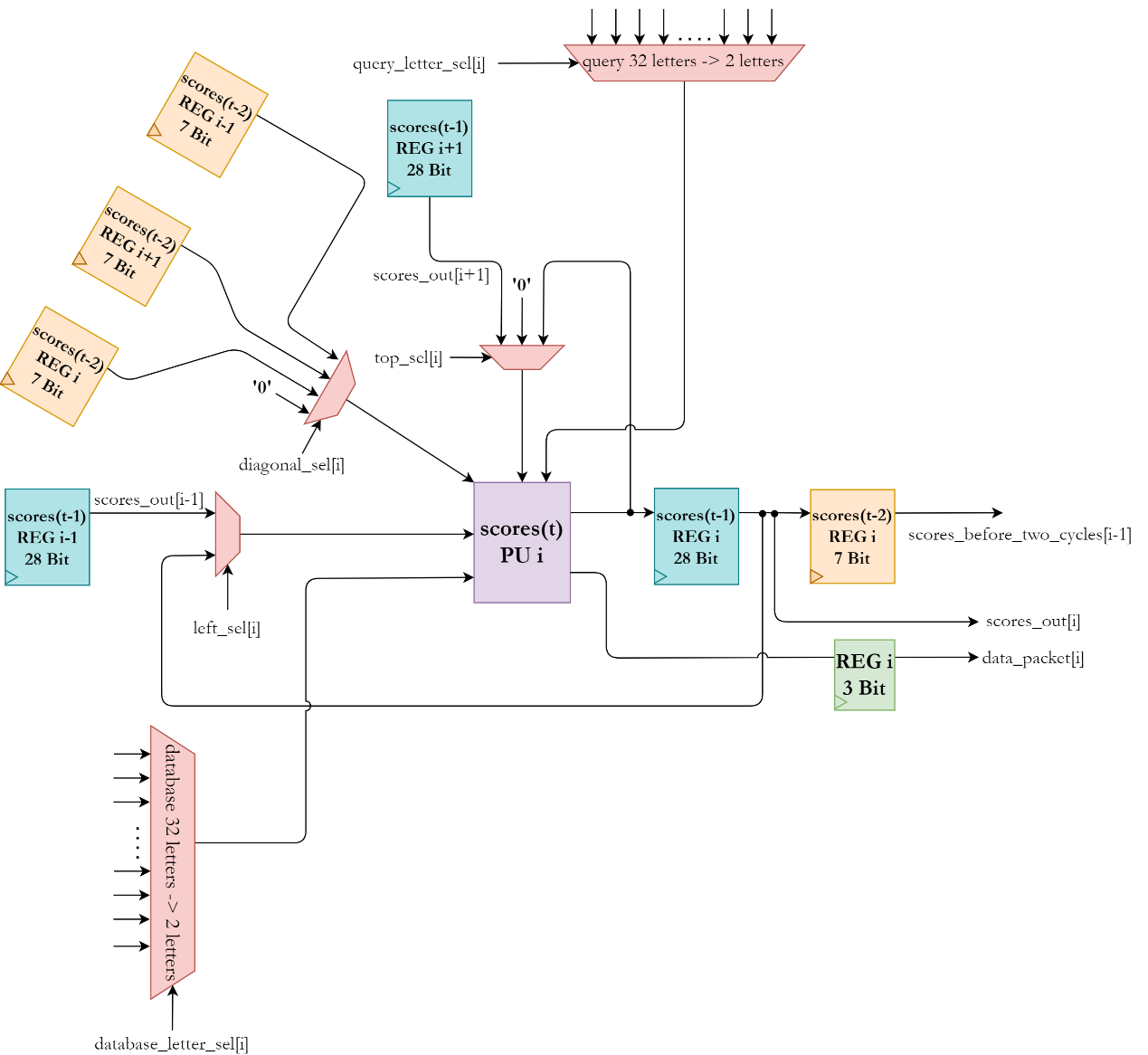


Figure 18: Interaction Between PUs

For the two adjacent top cells, it needs to choose between:

* Scores calculated by PUi in the previous cycle.
* Scores calculated by PUi+1 in the previous cycle.
* Constant zero scores (in case of the first row in the matrix).

For the two adjacent left cells, it needs to choose between:

* Scores calculated by PUi in the previous cycle.
* Scores calculated by PUi-1 in the previous cycle.

For the adjacent diagonal cell, it needs to choose between:

* Scores calculated by PUi in the previous cycle.
* Scores calculated by PUi-1 in the previous cycle.
* Scores calculated by PUi+1 in the previous cycle.
* Constant zero scores (in case of the top left cell in the matrix).

These decisions are made by the 'top\_sel', 'left\_sel' and 'diagonal\_sel' control signals. It is also necessary to determine the query and database letters related to the current cells that are computed by the PU. This is done by the 'query\_letter\_sel' and 'database\_letter\_sel' control signals. Explanation on the algorithm that determines the selectors' values is described in section 11.4.

1. **Matrix Memory**

תמונה שמכילה צילום מסך, עיצוב

התיאור נוצר באופן אוטומטיThe Matrix Memory Unit stores the data packets of all the matrix cells that computed by the Matrix Calculation Unit. To reduce area, the unit stores the source and the zero-score bit of each cell only, instead of keeping the whole score. The unit designed in a coned structure, for fast and efficient write to the registers.

Figure 19: Matrix Memory Unit

* 1. **Write Mechanism**

After every cycle of matrix computation, all the data packets related to one diagonal in the matrix are sent from the Matrix Calculation Unit to the Matrix Memory Unit. Each data packet input is connected to the input of the appropriated register of each diagonal and written only to one enabled register, which determined by the 'write\_ctl' signal provided by the controller. This is a one hot vector – the place in which the '1' is set represents the diagonal that will be enabled. When all the bits are '0', the write to the memory is disabled.

* 1. **Read Mechanism**

The read mechanism consists of 3 muxes. The first mux chooses all the scores of a specific diagonal, the second mux chooses the scores of a PU in that diagonal, and the third mux choose the requested data packet inside that PU. These 3 muxes are controlled by the following control signals:

1. choose\_diagonal – determines the diagonal with the required information out of the 31 diagonals in the matrix.
2. choose\_pu – determines the PU with the required information out of the 16 PUs.
3. choose\_pe – determines the PE with the required information out of the 4 PEs.

Those signals are necessary to send the required data packet to the Traceback Unit.

* 1. **Interface and Signals**

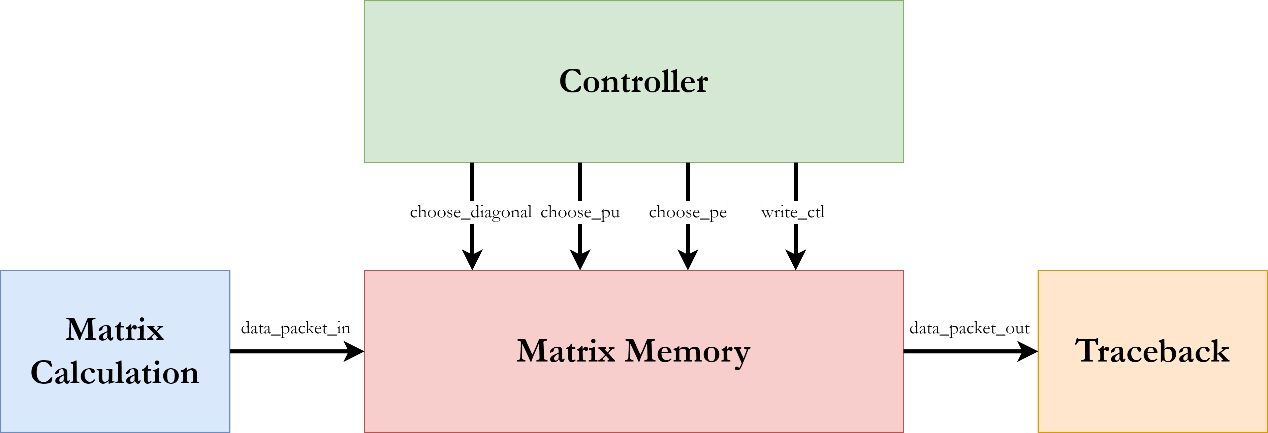


Figure 20: Matrix Memory Interface

The Matrix Memory Unit gets as an input all the data packets of the current diagonal from the Matrix Calculation Unit and outputs one data packet in a cycle for the Traceback Unit. It gets from the controller necessary control signals for reading the requested data packet for the Traceback Unit, as well as enable signals for writing the data packets from the Matrix Calculation Unit.

|  |  |  |  |
| --- | --- | --- | --- |
| **Signal Name** | **In/Out** | **Size (bits)** | **Description** |
| clk | input | 1 |  |
| rst\_n | 1 |  |
| write\_ctl | 31 | One hot vector that enables writing to the appropriate diagonal. |
| choose\_diagonal | 5 | Determines the requested diagonal for read. |
| choose\_pu | 4 | Determines the requested PU for read. |
| choose\_pe | 2 | Determines the requested data packet for read. |
| data\_packet\_in | 16x2x2x3 | Data packets of a single diagonal (consists maximum of 64 data packets). |
| data\_packet\_out | output | 3 | Data packet for the Traceback Unit. |

Table 12: Matrix Memory I/O Signals

1. **Max Registers**

The Max Registers Unit is responsible for finding the maximum cell in the Matrix. It is doing so by comparing all scores of a diagonal computed by the Matrix Calculation Unit in the previous cycle. To simplify the calculation, the computation is divided into 3 main stages:

1. Finding the maximum cell in each PU separately. There are up to 16 PUs computed by the Matrix Calculation Unit in one cycle, thus, there are 16 components that computes the maximum of each PU in parallel.
2. Finding the maximum cell in the current diagonal. It is computed by comparing the maximum cells computed in the previous stage and finding the maximum cell out of them.
3. Comparing the maximum cell of the current diagonal with the maximum cell already written into the registers. If the new score of the maximum cell is bigger than the score already written, the registers will be updated with the new maximum cell.

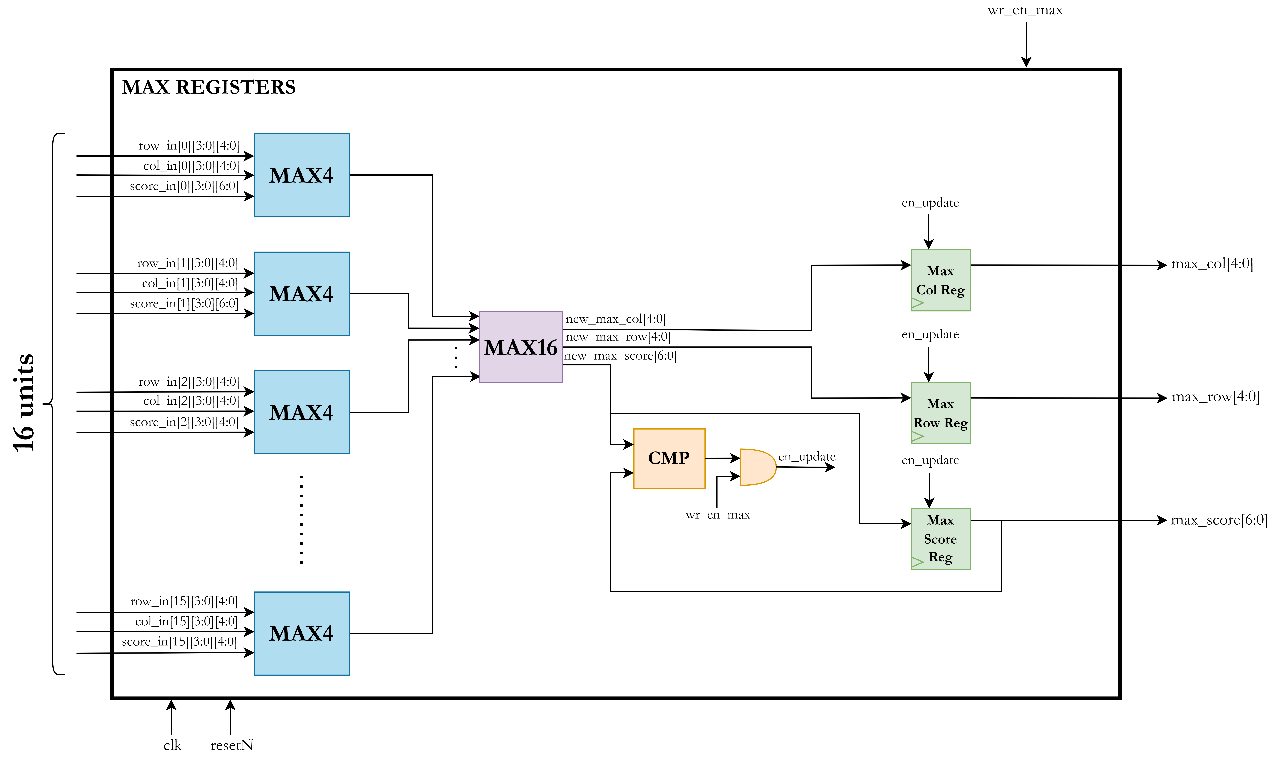
To perform this operation and update the maximum cell on each cycle, it is necessary that all these 3 stages will be performed in a single cycle. It is also necessary to keep the row and column indexes of the maximum cell, in addition to the score itself.

Figure 21: Max Registers Scheme

* 1. **Interface and Signals**

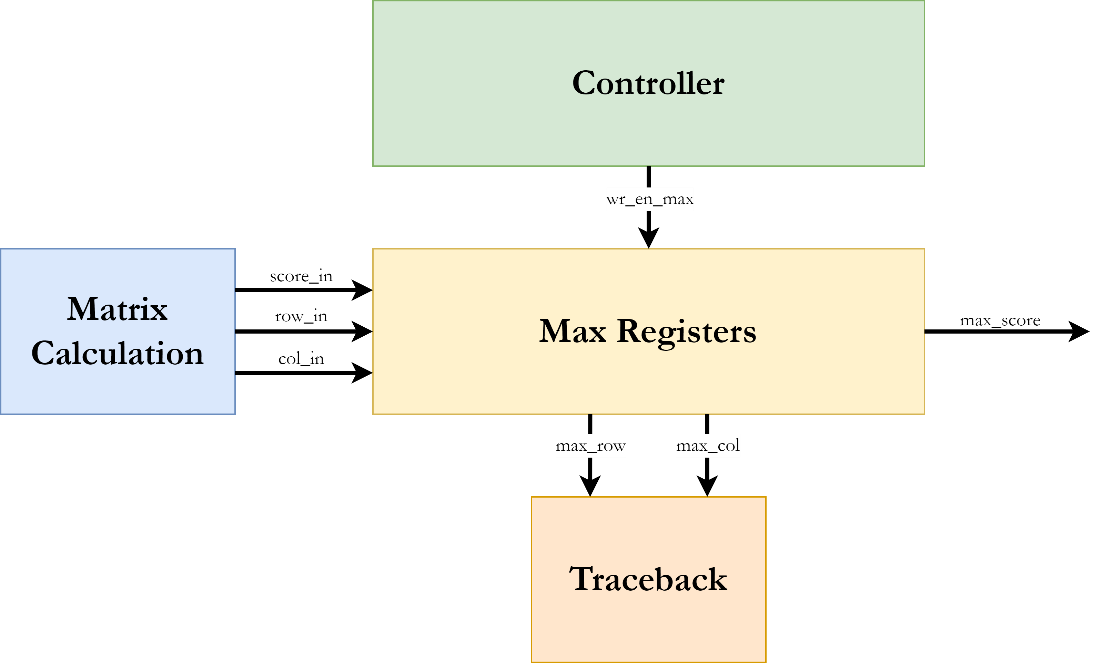


Figure 22: Max Registers Interface

The Max Registers Unit gets as an input the required information on each cell from the Matrix Calculation Unit. The indexes of the row and column of the maximum cell are sent to the Traceback Unit after the whole matrix calculation stage is finished. The maximum score is sent directly to the design output.

|  |  |  |  |
| --- | --- | --- | --- |
| **Signal Name** | **In/Out** | **Size (bits)** | **Description** |
| clk | input | 1 |  |
| rst\_n | 1 |  |
| row\_in | 16x4x5 | Row index of the cells computed in the previous cycle. |
| col\_in | 16x4x5 | Column index of the cells computed in the previous cycle. |
| score\_in | 16x4x7 | Scores of the cells computed in the previous cycle. |
| wr\_en\_max | 1 | Enable signal from the controller. |
| max\_score | output | 7 | The score of the maximum cell. |
| max\_row | 5 | The row index of the maximum cell. |
| max\_col | 5 | The column index of the maximum cell. |

Table 13: Max Registers I/O Signals

1. **Traceback**

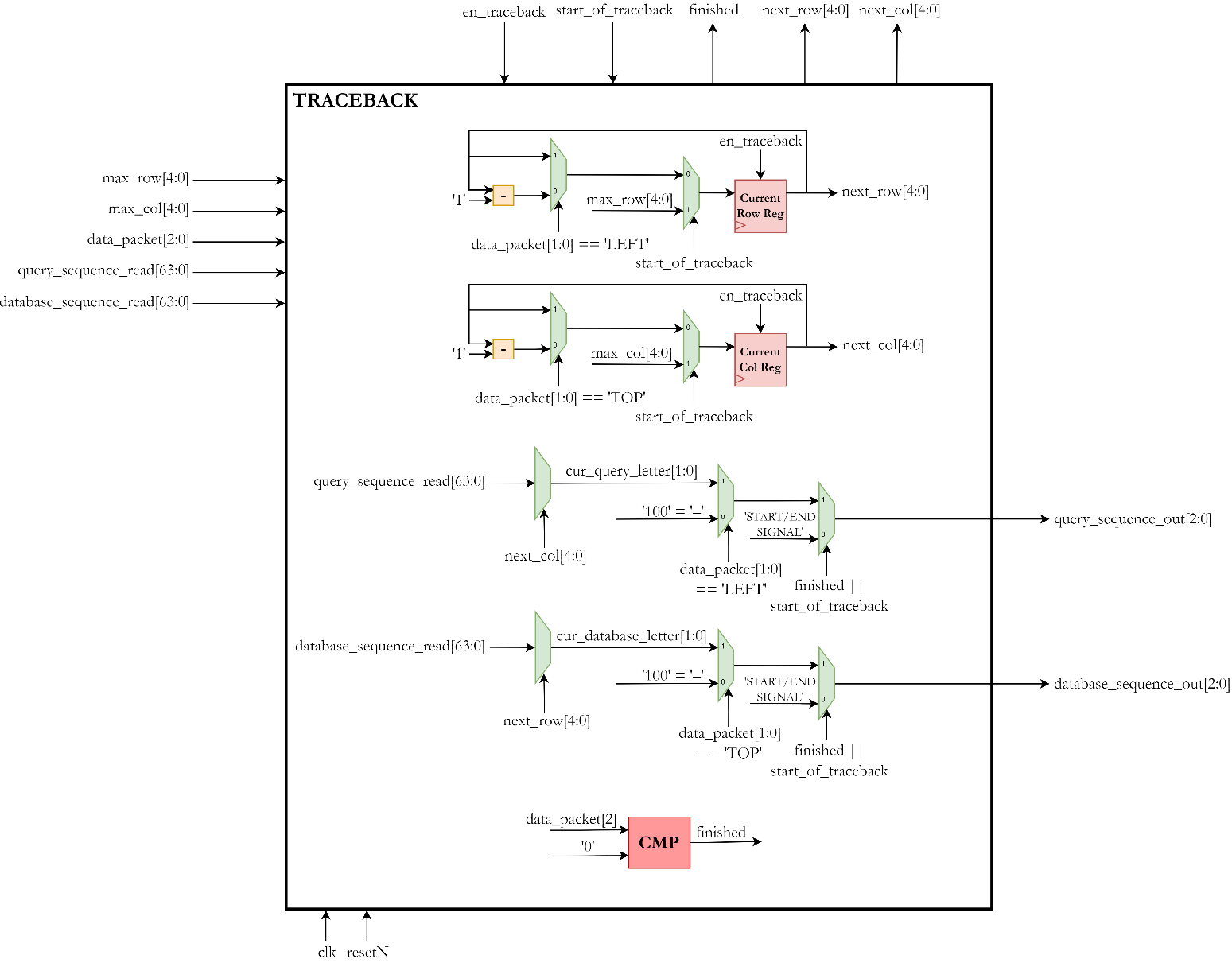
The Traceback Unit is responsible for output the aligned sequences. The unit does one step of trace back every cycle and outputs one letter from each sequence. Thus, the output of the aligned sequences is sent out in a reverse order.

Figure 23: Traceback Scheme

The traceback process starts when the unit gets from the controller 'start\_of\_traceback' pulse, and the registers are enabled by 'en\_traceback' signal. The unit starts in the cell with the highest score in the substitution matrix, and sends to the controller the indexes of the row and column for the next traceback step. On every cycle, the unit calculates the next query and database letters of the aligned sequences, in parallel to the next row and next column calculation. This process continues until the current cell has a zero score bit that equals zero.

* 1. **Interface and Signals**

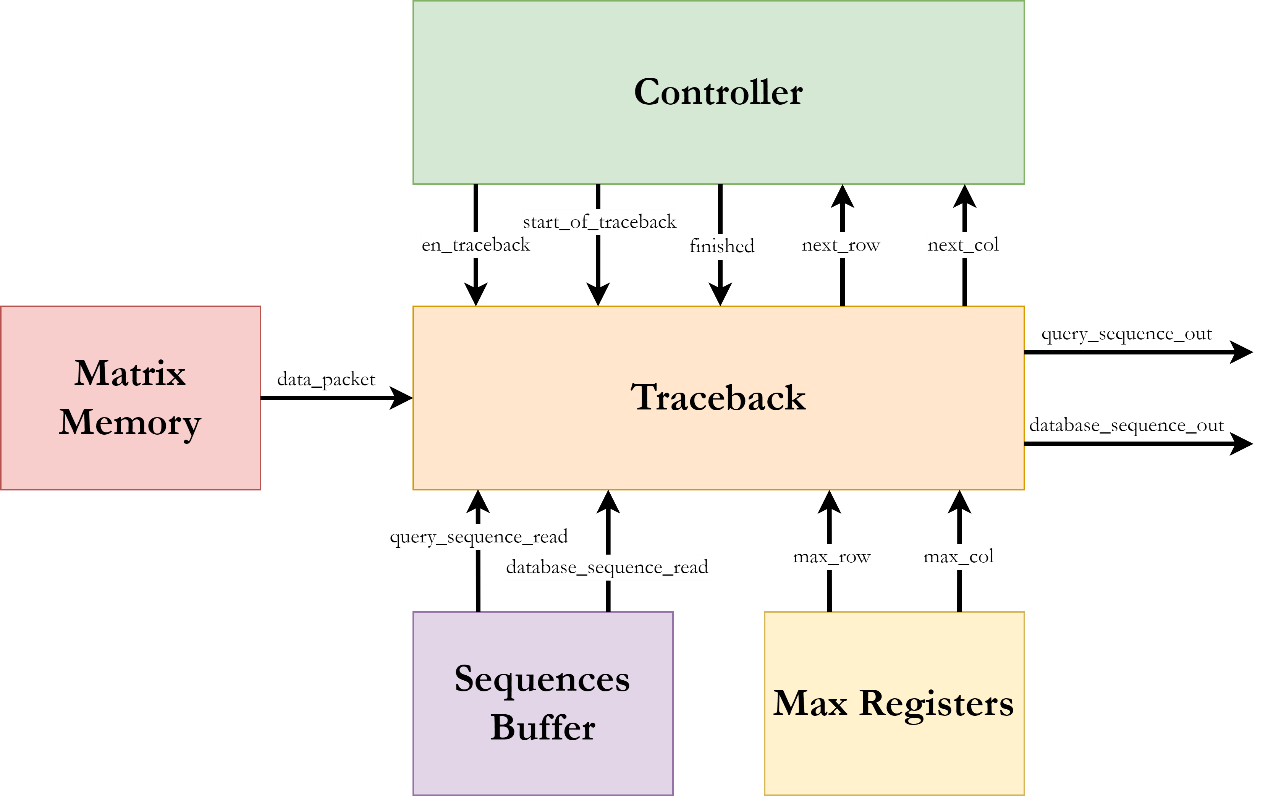


Figure 24: Traceback Interface

The Traceback Unit gets the information about the maximum cell in the matrix from the Max Registers Unit. It gets on each cycle the information of the current cell in traceback process from the Matrix Memory Unit and the query and database sequences from the Sequences Buffer Unit.

|  |  |  |  |
| --- | --- | --- | --- |
| **Signal Name** | **In/Out** | **Size (bits)** | **Description** |
| clk | input | 1 |  |
| rst\_n | 1 |  |
| en\_traceback | 1 | Enable signal for the unit's registers. |
| start\_of\_traceback | 1 | Indicates on start of the traceback stage. |
| finished | 1 | Indicate to the controller that the traceback stage has finished. |
| next\_row | 5 | The row index of the next cell in the traceback process. |
| next\_col | 5 | The column index of the next cell in the traceback process. |
| max\_row | 5 | The row index of the maximum cell in the substitution matrix. |
| max\_col | 5 | The column index of the maximum cell in the substitution matrix. |
| data\_packet | 3 | The data packet of the current cell in the traceback process. |
| query\_sequence\_read | 2 | Current query letter. |
| database\_sequence\_read | 2 | Current database letter. |
| query\_sequence\_out | output | 3 | Current query letter of the aligned sequence. |
| database\_sequence\_out | 3 | Current database letter of the aligned sequence. |

Table 14: Matrix Memory I/O Signals

1. **Controller**

The controller operates as a Finite State Machine, interacting with all the units within the chip. Its role is to send enable and control signals to all the sub-units and to manage communication between sub-units. It is responsible to manage the data flow across the entire system.

* 1. **Controller as a FSM**

The controller is a finite state machine implemented as a mealy machine.

תמונה שמכילה טקסט, צילום מסך, עיגול

התיאור נוצר באופן אוטומטי

Figure 25: Controller's Finite State Machine

The controller is maintained by the 'global\_counter' internal signal. Global counter is a 5-bit counter which counts from 0 to 31. It is responsible for switching between states depending on its current value. Global counter is active only when 'busy' signal is active and when the FSM is not in 'IDLE\_ST'. When 'finished' signal is received, the counter value is set to 0.

The Controller's FSM has 4 states:

* **IDLE\_ST** – This state indicates that the chip is currently does not working on a sequence alignment task. Only when a "handshake" is made: the controller asserts 'ready' signal and the user send 'start' signal for a single pulse, the FSM moves to the next state.
* **LOAD\_SEQUENCES\_ST –** In this state the global counter starts to count. Furthermore, the operation of the Sequences Buffer Unit is enabled. When global counter 1,Matrix Calculation Unit is enabled and when global counter 2,Max Registers Unit and Matrix Memory Unit are enabled.This state takes exactly 8 cycles.
* **SCORES\_CALC\_ST –** This state takes 24 cycles exactly, during which the substitution matrix is continued to be filled.
* **TRACEBACK\_ST –** In this state the Traceback Unit start to perform until a 'finished' signal is sent to the controller. The global counter is inactive during this state.
  1. **Controller Interface**

The controller has an interface with the following units:

1. Matrix Calculation
2. Sequence Buffer
3. Max Registers
4. Matrix Memory
5. Traceback

תמונה שמכילה טקסט, צילום מסך, גופן

התיאור נוצר באופן אוטומטי

Figure 26: Controller Interface

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Interface** | **Signal** | **In/Out** | **Size (bits)** | **Description** |
| Clk & Reset | clk | input | 1 |  |
| rst\_n | 1 |  |
| Sequence Buffer | wr\_en\_buff | output | 1 | Sequences Buffer enable. |
| count\_buff | 3 | Sequences Buffer counter. |
| Matrix Calculation | top\_sel | output | 16x2 | Selector for the top values in the PUs array. |
| left\_sel | 16x2 | Selector for the left values in the PUs array. |
| diagonal\_sel | 16x2 | Selector for the diagonal value in the PUs array. |
| query\_letter\_sel | 16x2x5 | Selector for the query letters in the PUs array. |
| database\_letter\_sel | 16x2x5 | Selector for the database letters in the PUs array. |
| wr\_en\_pu | 16 | Scores out enable. |
| Max Registers | wr\_en\_max | output | 1 | Max Registers enable. |
| row\_in | 16x4x5 | Row indexes of the current diagonal cells. |
| col\_in | 16x4x5 | Column indexes of the current diagonal cells. |
| Matrix Memory | write\_ctl | output | 31 | One hot vector that enables write for the appropriate registers of the current diagonal. |
| choose\_diagonal | 5 | Requested diagonal for read. |
| choose\_pu | 16 | Requested PU for read. |
| choose\_pe | 2 | Requested PE for read. |
| Traceback | en\_traceback | output | 1 | Traceback enable. |
| start\_of\_traceback | 1 | Indicates to start the traceback process. |
| next\_row | input | 5 | The row index of the next cell in the traceback process. |
| next\_col | 5 | The column index of the next cell in the traceback process. |
| finished | 1 | Indication for finishing traceback process. |
| Output | output\_valid | output | 1 | Indicates output is valid. |

Table 15: Controller I/O Signals

* + 1. **Sequences Buffer**

The controller provides to the Sequences Buffer Unit an enable signal called 'wr\_en\_buff' to allow its use and a counter signal called 'count\_buff' that enables writing into the right registers of the Sequences Buffer.

* + 1. **Matrix Calculation**

The Controller provides control signals for all the 16 Processing Units of the Matrix Calculation Unit. The controller ensures that the Matrix Calculation unit selects the right values for each Processing Unit during every computation cycle. Here is a pseudo code describes the algorithm of generating the control signals:

Query letters select:

for i in range (0, NUM\_PU):

if (global\_counter <= 16): # Upper half

query\_letter\_sel[i][0] = 2\*j

query\_letter\_sel[i][1] = 2\*j + 1

else: # Lower half

query\_letter\_sel[i][0] = 2\*(global\_counter - 1 + i) - 30

query\_letter\_sel[i][1] = 2\*(global\_counter - 1 + i) – 30 + 1

Database letters select:

for i in range (0, NUM\_PU):

if (global\_counter <= 16): # Upper half

database\_letter\_sel[i][0] = 2\*(global\_counter - 1 – i)

database\_letter\_sel[i][1] = 2\*(global\_counter - 1 – i) + 1

else: # Lower half

database\_letter\_sel[i][0] = 31 - (2\*i + 1)

database\_letter\_sel[i][1] = 31 - 2\*i

Top scores select:

for i in range (0, NUM\_PU - 1):

if (global\_counter <= 16):

if (i == 0):

if (global\_counter == 1):

top\_sel[i] = 0

else:

top\_sel[i] = 1

else if (i == global\_counter – 1):

top\_sel[i] = 0

else:

top\_sel[i] = 1

else:

top\_sel[i] = 2

Left scores select:

for i in range (0, NUM\_PU - 1):

if (global\_counter <= 16): # Upper half

if (i == 0): # PU number 0

left\_sel[i] = 0

else:

left\_sel[i] = 1

else: # Lower half

if (i == 0): # PU number 0

left\_sel[i] = 1

else:

left\_sel[i] = 0

Diagonal scores select:

for i in range (0, NUM\_PU - 1):

if (global\_counter <= 16): # Upper half

if (i == 0): # PU number 0

diagonal\_sel[i] = 0

else if (i == global\_counter – 1): # PUs on the first row

diagonal\_sel[i] = 0

else: # Lower half

diagonal\_sel[i] = 2

else:

if (global\_counter == 17): # Main diagonal

diagonal\_sel[i] = 1

else:

if (i == 0): # PU number 0

diagonal\_sel[i] = 2

else:

diagonal\_sel[i] = 3

* + 1. **Max Registers**

The controller provides the Max Registers Unit an enable signal called 'wr\_en\_max' that allows it to operate. In addition, two vectors 'row\_in' and 'col\_in' represent the indexes of the cells in the current computation cycle of the PUs array.

* + 1. **Matrix Memory**

The controller generates control signals for read and write operations in the Matrix Memory Unit. These signals specify the registers for reading or writing the data. Here is a pseudo code describes the algorithm of generating the control signals:

Choose diagonal:

choose\_diagonal = (next\_row >> 1) + (next\_col >> 1)

Choose PU:

if (next\_row + next\_col 31):

choose\_pu = next\_col >> 1

else:

choose\_pu = (31 – next\_row) >> 1

Choose PE:

if (next\_row[0] == 1) && (next\_col[0] == 1):

choose\_pe = 2'b11; # bottom right PE

else if (next\_row[0] == 0 && next\_col[0] == 0):

choose\_pe = 2'b00; # top left PE

else if (next\_row[0] == 0 && next\_col[0] == 1):

choose\_pe = 2'b01; # top right PE

else if (next\_row[0] == 1 && next\_col[0] == 0):

choose\_pe = 2'b10; # bottom left PE

* + 1. **Traceback**

The controller manages the Traceback Unit by transmitting control signals and additionally mediates the communication with the Matrix Memory Unit. The controller provides a single pulse called 'start\_of\_traceback' when it needs to start the traceback process. During the traceback stage, a signal called 'en\_traceback' is set to '1', enabling the registers in the unit. When the traceback stage is done, a 'finished' signal is sent to the controller.

The Traceback Unit sends to the controller two signals: 'next\_row' and 'next\_col'. The controller translates them into select signals for reading the correct data from the Matrix Memory Unit.

1. **Simulations and Verification**

An important part of the design process is performing a verification of the chip. To avoid bugs, after the implementation of each unit is completed, we checked its behavior using logic simulations and verified that the signals seen in the simulation are corresponding to the expected behavior.

In this chapter will be presented a small subset of the simulations we made. The simulations made using Verdi debug and verification management platform of Synopsys.

* 1. **Sequences Buffer**

תמונה שמכילה צילום מסך, קו, מחשב

התיאור נוצר באופן אוטומטי

Figure 27: Sequences Buffer Simulation

The first 4 query letters received as an input on the first cycle are 10011100, which is TGCA. These letters are seen in the output after they are stored in the registers on the next clock rise. Similarly, the first 4 database letters received as an input on the first cycle are 10111011, which is TCTC. These letters are seen in the output after they are stored in the registers on the next clock rise. On the second cycle, the next 4 letters from each sequence are stored in the next registers, and so on.

* 1. **Processing Element**

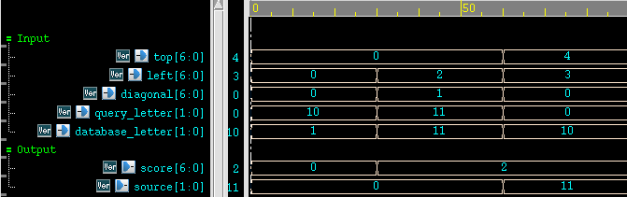


Figure 28: Processing Element Simulation

On the first cycle, all the 3 input scores are equal to 0 and there is a mismatch. Thus, the output score is 0 and the source is 00 (diagonal) by default. On the second cycle, the top score is 0, the left score is 2, the diagonal score is 1 and there is a match. Thus, the output score is 2 and the source is 00 (diagonal). On the third cycle, the top score is 4, the left score is 3, the diagonal score is 0 and there is a mismatch. Thus, the output score is 2 and the source is 11 (top).

* 1. **Processing Unit**

תמונה שמכילה צילום מסך, קו

התיאור נוצר באופן אוטומטי

Figure 29: Processing Unit Simulation

Figure 30 shows the expected output related to the given simulation inputs.

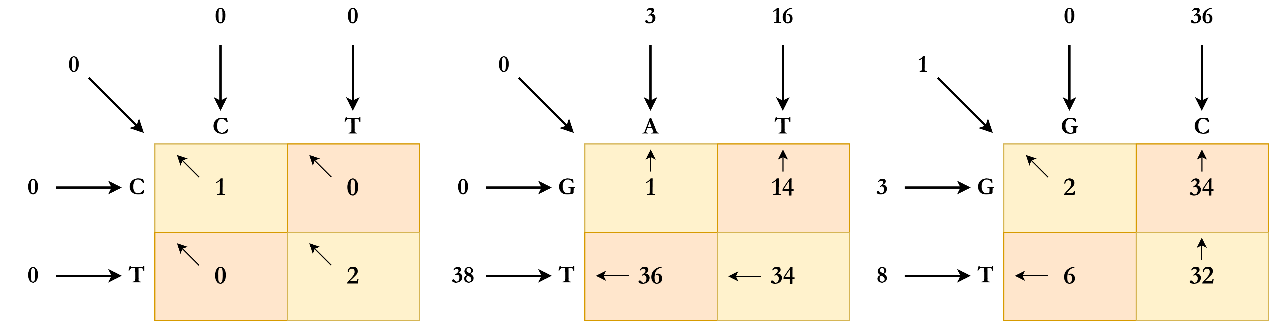


Figure 30: Processing Unit Expected Output

The results shown in the simulation correspond to the expected behavior of the PU unit.

* 1. **Matrix Calculation**
  2. **Matrix Memory**
  3. **Max Registers**
  4. **Traceback**
  5. **Full Simulations**

1. **Summary and Conclusions**

This project introduced a parallel implementation of Smith-Waterman Algorithm by a hardware accelerator. It has been a great milestone in our way to become electrical engineers.

Before start working on this project, the Sequence Alignment Problem was unfamiliar for us. We had to dive into a new topic and to understand comprehensively the algorithm. We learned during this project how to plan a complex system, different approaches of chips architecture, tradeoffs, and limitations we must consider, System Verilog skills, and much more. We also learned how to solve problems independently and how much important is preliminary planning.

We would like to express our sincere appreciation to the VLSI lab stuff, especially to our instructor, Goel Samuel. His invaluable assistance played a pivotal role in the realization of our project. We are also thankful to Apple Israel for their collaboration and generous funding of this project, and especially to Dalia Haim, who is leading this collaboration. We are extending our gratitude to everyone who contributed to the project's success.

This project has been a remarkable experience of learning, development, and expanding horizons. We are looking forward to continuing with the sequel project and to see our chip working!

1. **References**
2. Eddy, S. R. (2008). A probabilistic model of local sequence alignment that simplifies statistical significance estimation. *PLoS computational biology*, *4*(5), e1000069.‏
3. Liu, J. S., Neuwald, A. F., & Lawrence, C. E. (1995). Bayesian models for multiple local sequence alignment and Gibbs sampling strategies. *Journal of the American statistical Association*, *90*(432), 1156-1170.‏
4. Roĭtberg, M. A. (2004). Comparative analysis of primary structure of nucleic acids and proteins. *Molekuliarnaia Biologiia*, *38*(1), 92-103.