

در این آزمایش، ما یک شیفت دهنده 4 بیتی با ورودی reset در منطق منفی که 4 حالت (SHR, SHL, SAR, SAL) دارد و یک تقسیم کننده 8 بیت بر 4 بیت را پیاده سازی کردیم.

شیفت دهنده ما یک ورودی 2 بیتی دارد که حالت های مختلف آن نوع انجام شیفت را نشان می دهند که عبارتند از :

"00" : شیفت منطقی به سمت چپ که به اول آن بیت 0 اضافه می گردد.

"01" : شیفت منطقی به سمت راست که به آخر آن بیت 0 افزوده می گردد.

"10" : شیفت ریاضی به سمت چپ که به اول آن بیت 0 می افزاید.

"11" : شیفت ریاضی به سمت راست که بیت آخر آن تکرار می گردد (این حالت برای اعداد علامتدار کاربرد دارد برای جلوگیری از تغییر علامت عدد).

تقسیم کننده ما با استفاده از الگوریتمی که در کلاس آموزش داده شد پیاده سازی شده است که برای آن کلاک و reset در منطق منفی طراحی گردیده و همچنین با استفاده از حالت های مختلف پیاده سازی گشته است.

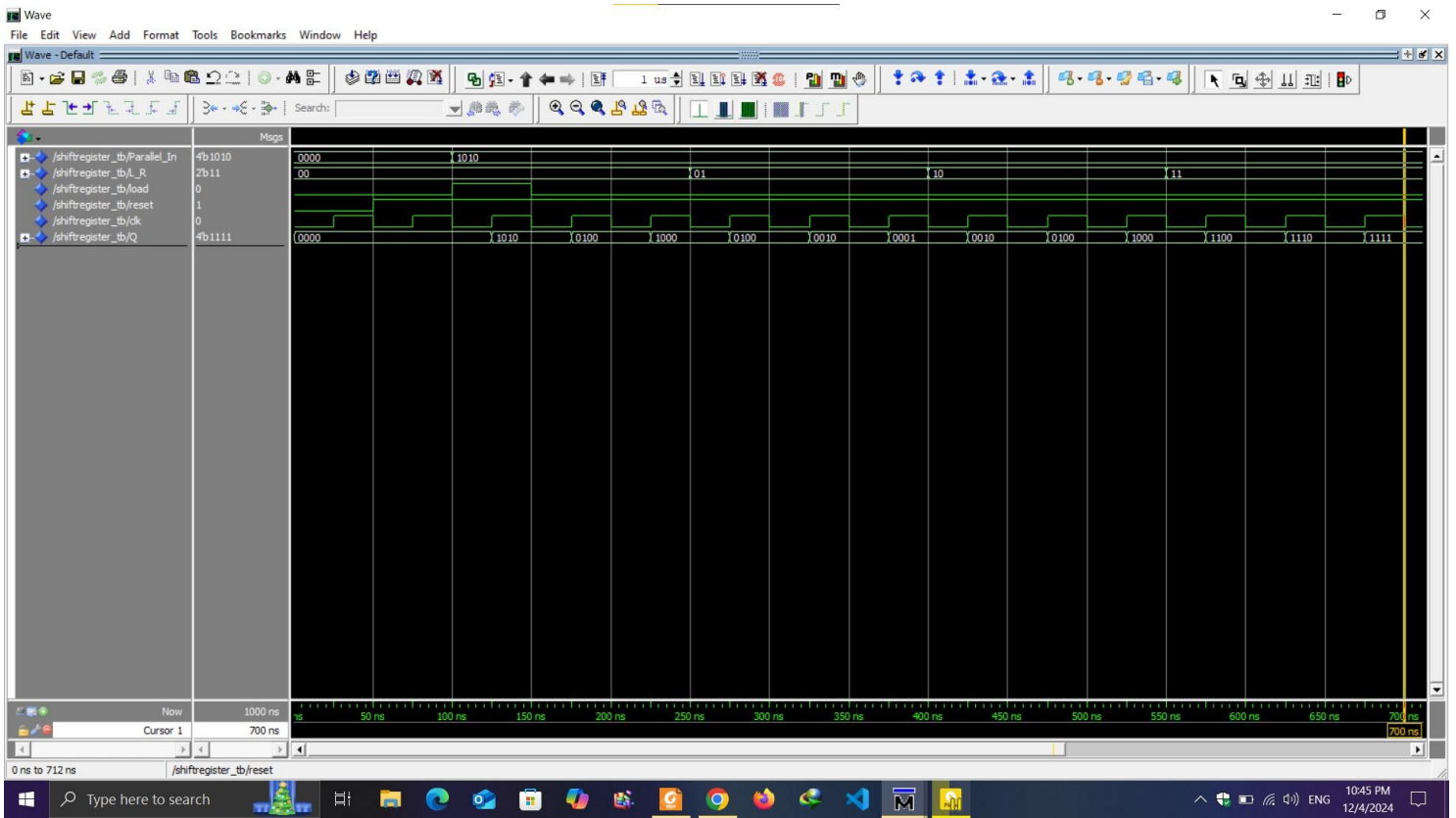
همچنین برای انجام تفریق آن از تفریق کننده مکمل گیر که فرمول آن به صورت $(A - B = A + B' + 1)$ است؛ استفاده نمودیم و برای انجام این عمل جمع نیز از Full Adder و Ripple Adder که 4 بیتی است؛ استفاده کرده و کد آنرا نوشتیم.

برای انجام این تقسیم، ابتدا ما عدد 8 بیتی را به دو قسمت 4 بیتی تقسیم کرده و هر بار این عمل تفریق مکمل گیر را روی این بخش بزرگ عدد و مقسوم علیه انجام می دهیم. اگر در شروع کار حاصل این تفریق یک بیت carry تولید کند؛ یعنی که تقسیم ما overflow دارد و نمیتوان آنرا انجام داد.

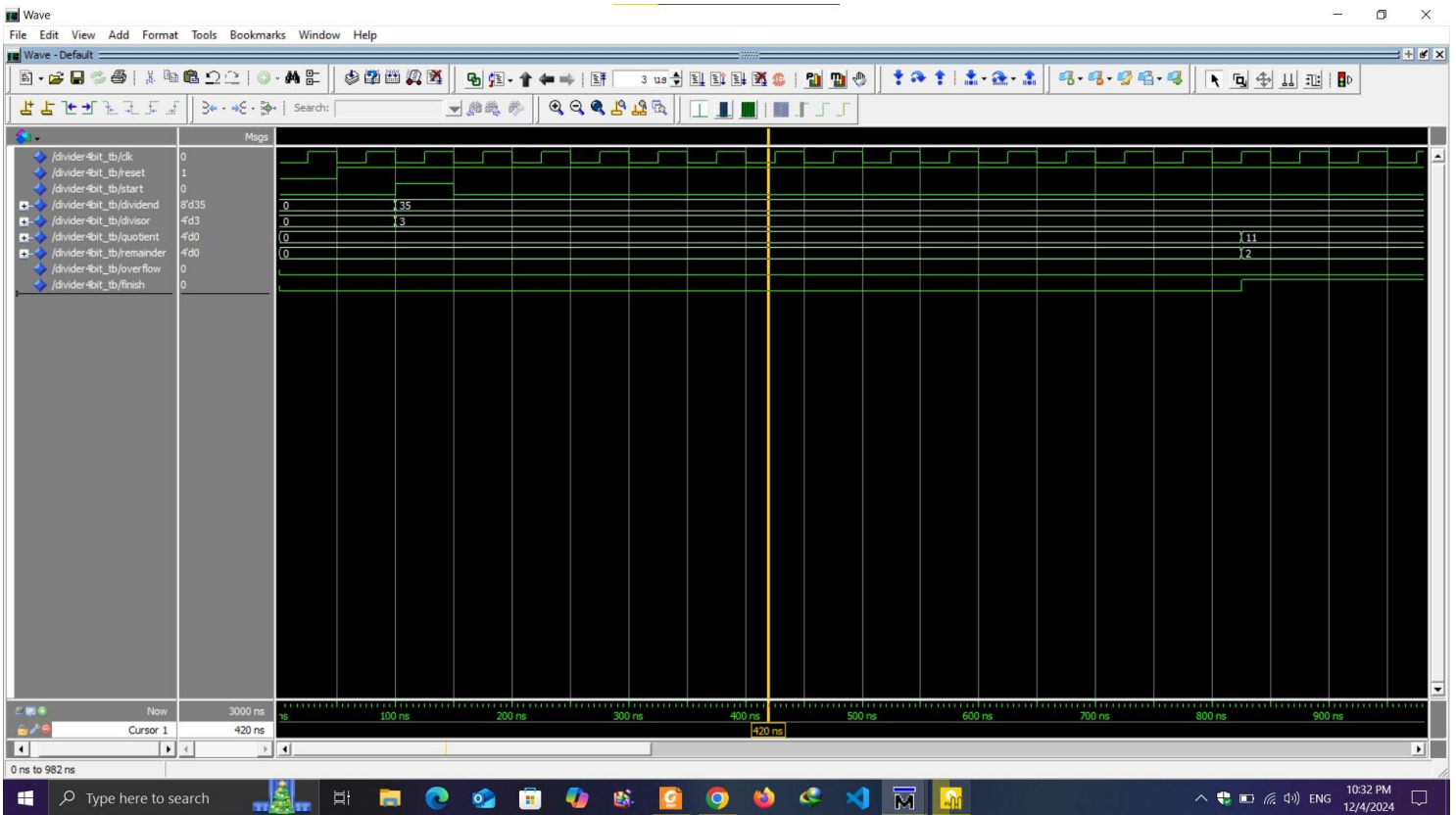
این فرایند تفریق اگر overflow نداشته باشد به اندازه بیت مقسوم علیه طول خواهد کشید که توجه کنید ممکن است بیشتر از تعداد کلاک باشد زیرا ملاک تمام شدن عملیات صفر شدن شمارنده ماست و نه تعداد کلاک؛ زیرا ما در هر کلاک در یک حالت مختلف هستیم که ممکن است حالت انجام تفریق نباشد.

جواب این تقسیم به صورت دو عدد 4 بیتی است که خارج قسمت و باقیمانده را نشان می دهد.

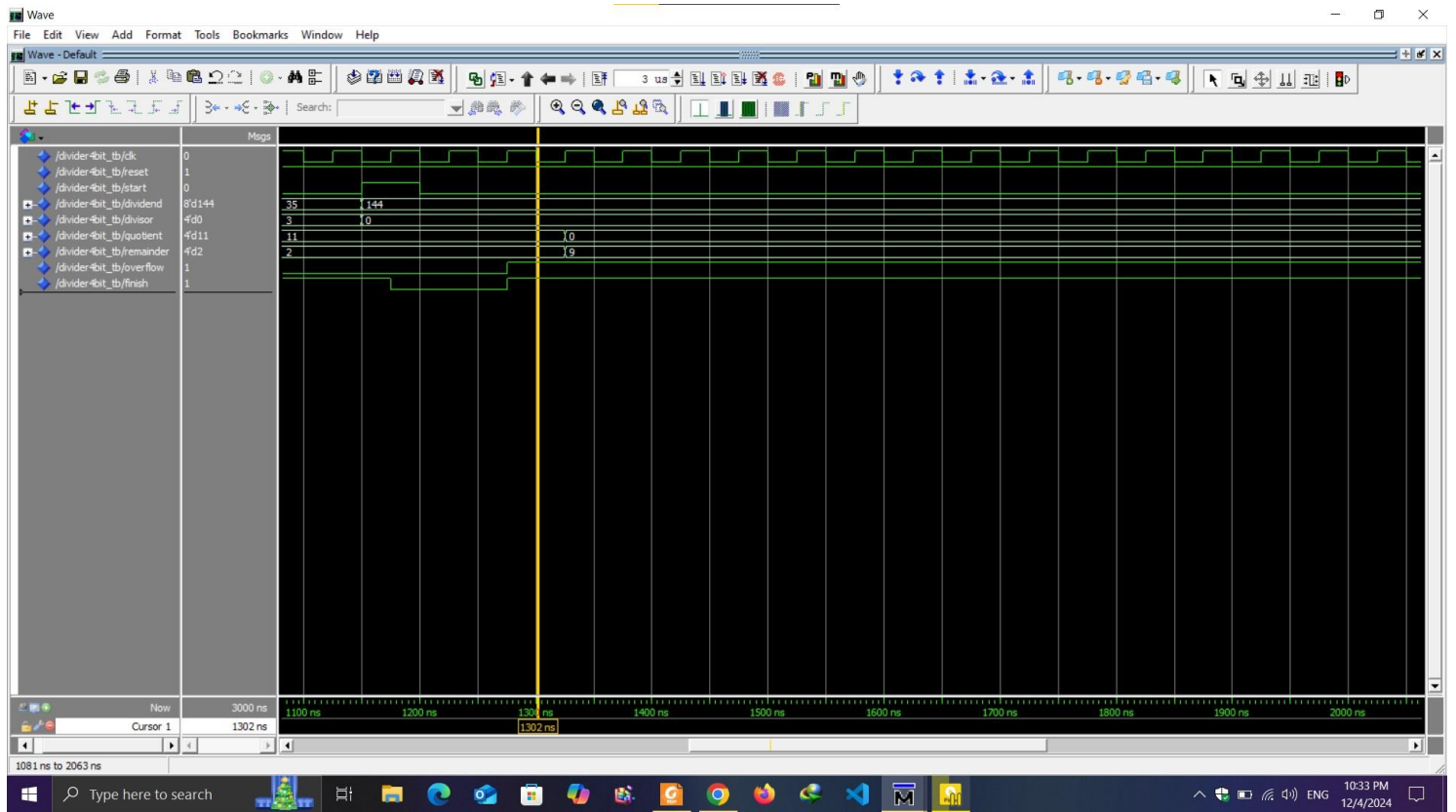
....در ادامه عکس از نتایج شبیه سازی های هر کدام از قسمت ها قرار داده شده است:



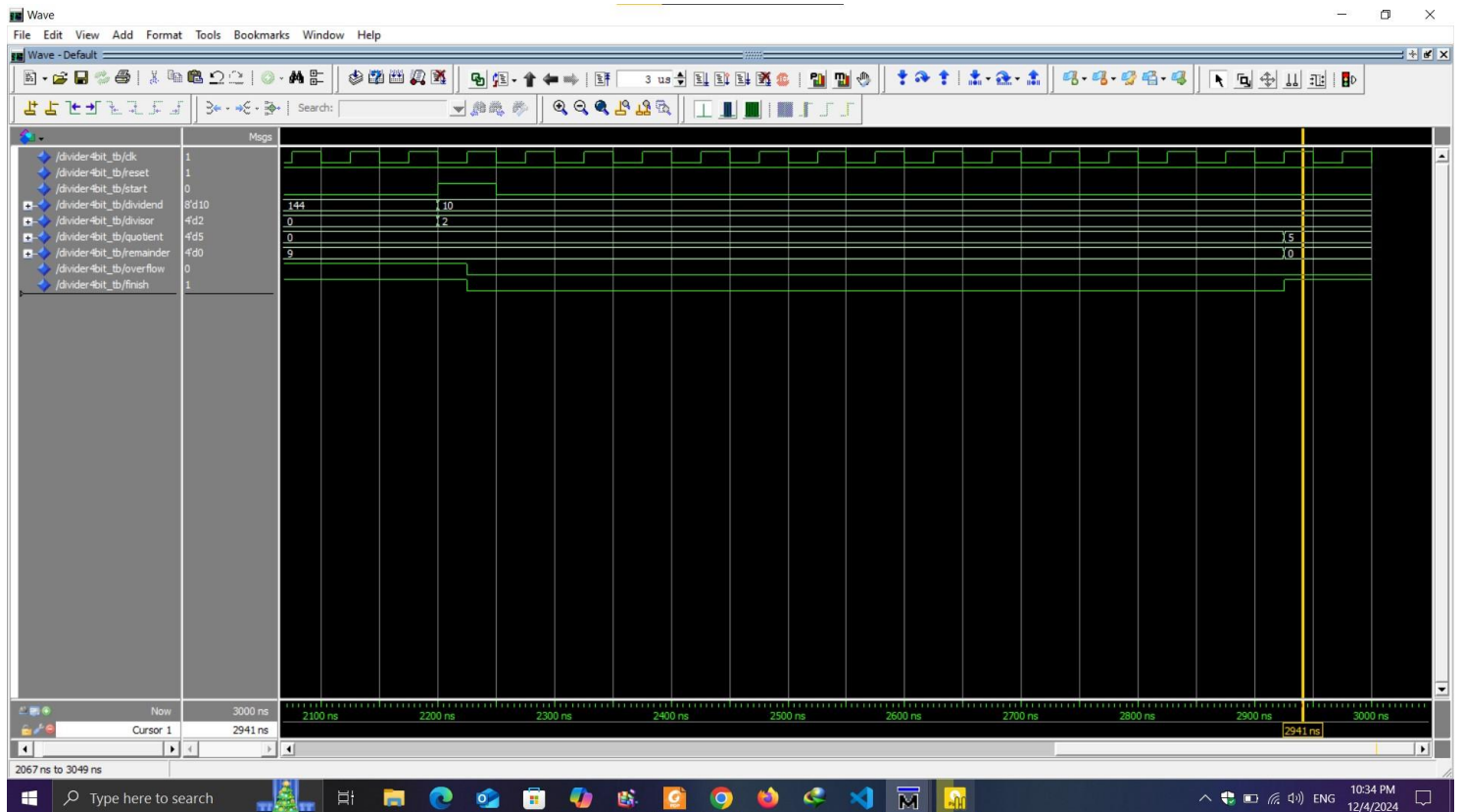
Shift Register 4 bit



Divider ($35/3 = 11, 2$)



Divider (144/0 = overflow)



Divider (10/2 = 5, 0)