

در این آزمایش، ابتدا ما باید برای توصیف ضرب کننده معمولی و ضرب کننده (Carry Save Adder) واحد های جمع کننده Half Adder و Full Adder را پیاده سازی کنیم.

همچنین برای پیاده سازی ضرب کننده ارایه ای نیز ما از (Ripple Adder) به عنوان واحد جمع کننده در هر لایه استفاده خواهیم کرد؛ پس انرا نیز پیاده سازی میکنیم.

برای **ضرب کننده معمولی** ما از همان الگوی داخل گزارش کار آزمایشگاه بهره برده و هر لایه را بر اساس ان پیاده سازی و مقادیر را با FA, HA محاسبه میکنیم تا در لایه اخر جواب ما بدست آید.

برای **ضرب کننده ارایه ای** نیز همان طور که قبلا گفتیم؛ از واحد جمع کننده RA در هر لایه ان استفاده میکنیم که کار ما را در اختصاص داده ها به جمع کننده و گرفتن داده ها بر خلاف قبلی اسانتر میکند و ما میتوانستیم از هر جمع کننده n-bit دیگری استفاده کنیم.

برای **ضرب کننده (Carry Save Adder)** ما می اییم و مانند ضرب کننده معمولی لایه ها را بر اساس الگوی داده شده در داخل گزارش پیاده سازی میکنیم ولی بر خلاف ضرب کننده معمولی، این ضرب کننده یک لایه بیشتر داشته به طوری که Carry را به مرحله بعد انتقال و از ان استفاده میکند ولی در تعداد واحد های FA, HA تفاوتی در میان این دو ضرب کننده نیست.

ضرب کننده بوث:

این ضرب کننده بر اساس مدل بوث عمل کرده و بر خلاف قبلی ها که ضرب کننده های بی علامت هستند؛ این ضرب کننده برای اعداد مکمل 2 کاربرد دارد و روی انها استفاده میگردد. همچنین این ضرب کننده مدار ترتیبی دارد بر خلاف قبلی ها پس مدتی برای تولید جواب طول میکشد که این مدت به اندازه n-clock میباشد.

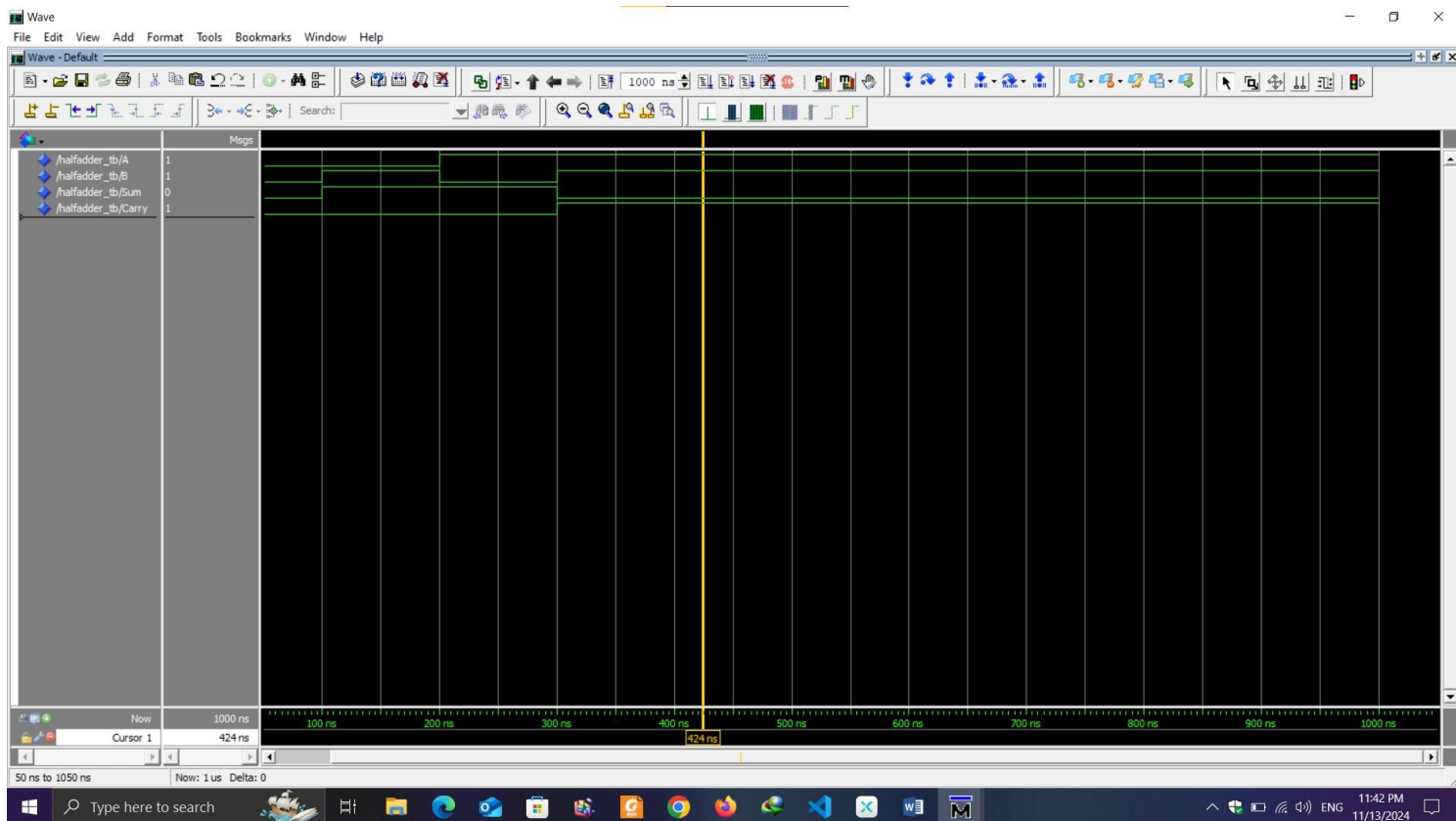
--حواسمان باشد که جواب ضرب کننده n-bit در 2n-bit ذخیره میشود.

--ضرب کننده های پیاده سازی شده همه ضرب کننده های 4 بیتی هستند

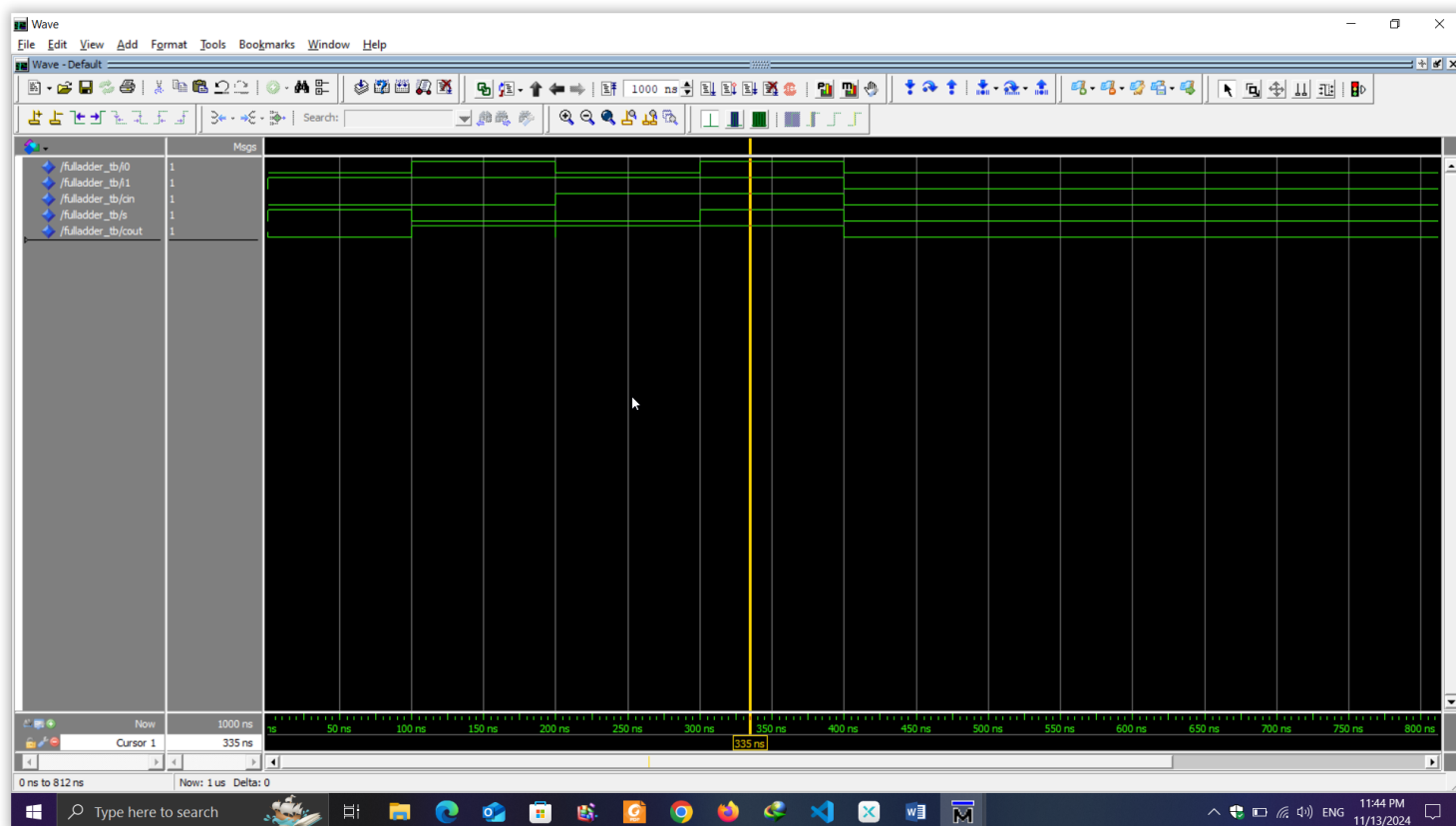
--رنج عدد از (0, 15) و رنج جواب از (0, 255) است.

--در ضرب کننده بوث به دلیل نوع نمایش مکمل 2، این رنج به صورت عدد از (7, -8) و جواب از (127, -128) است.

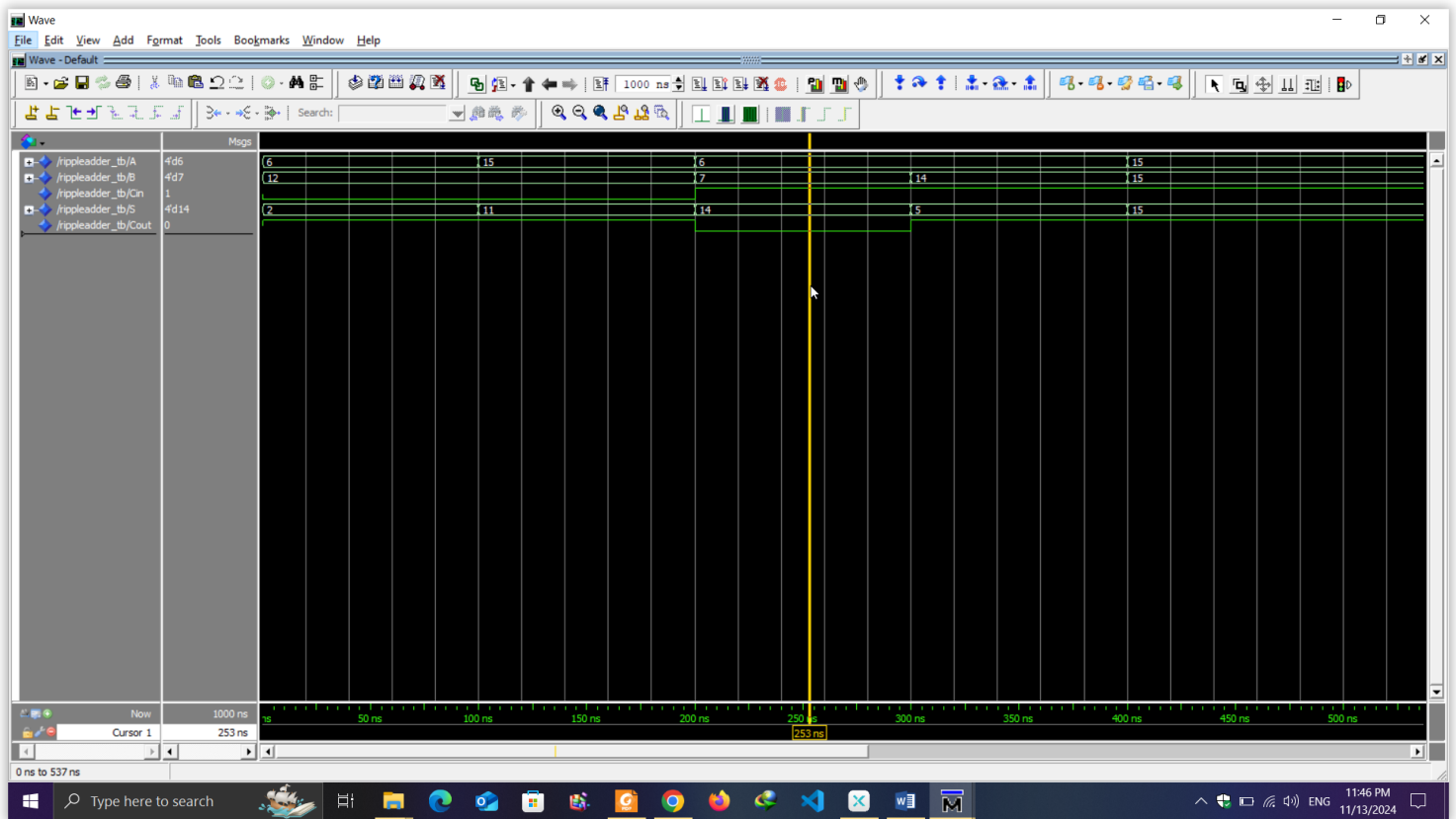
.....در ادامه عکس از نتایج شبیه سازی های هر کدام از گیت ها قرار داده شده است:



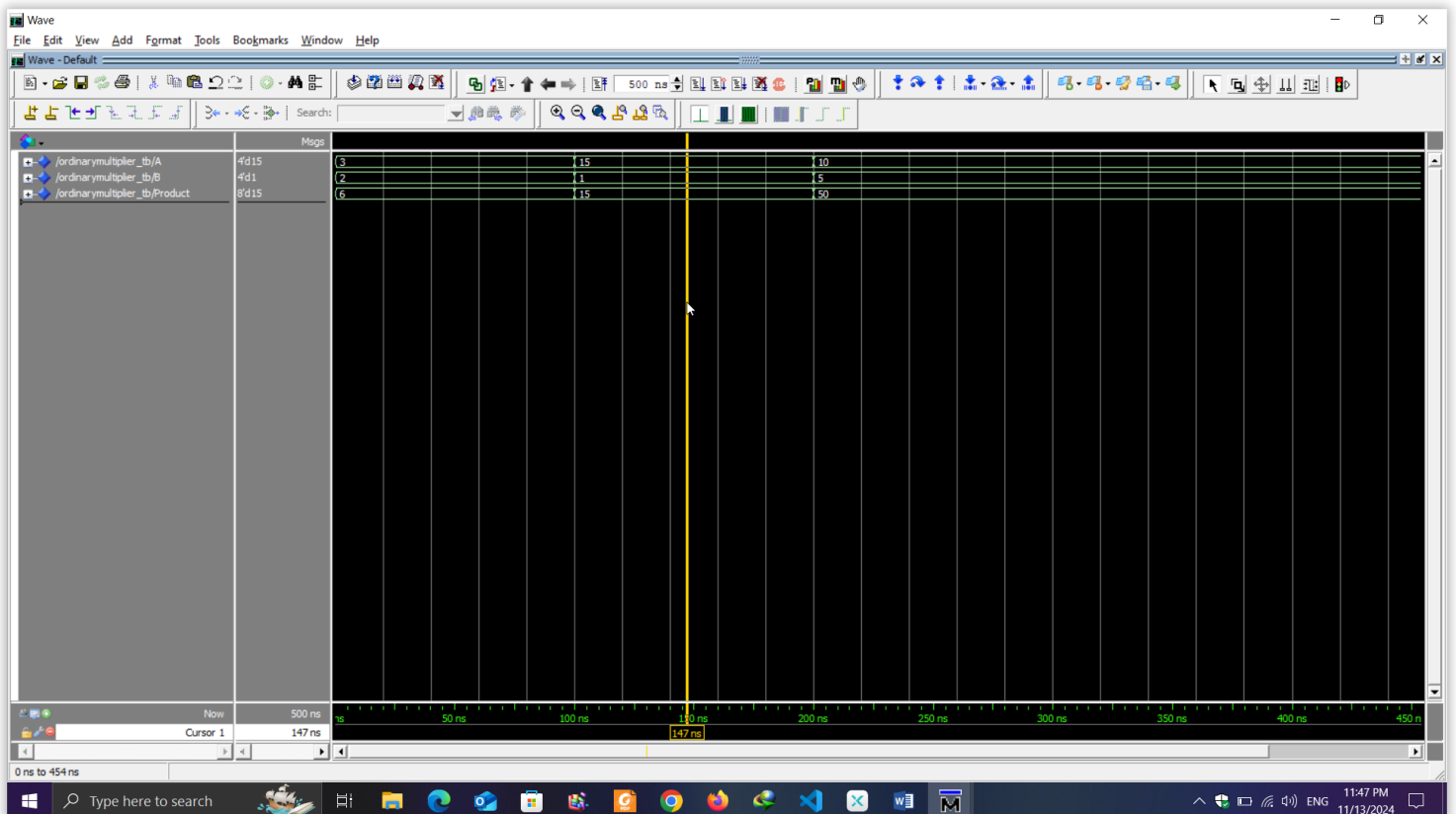
HF simulate



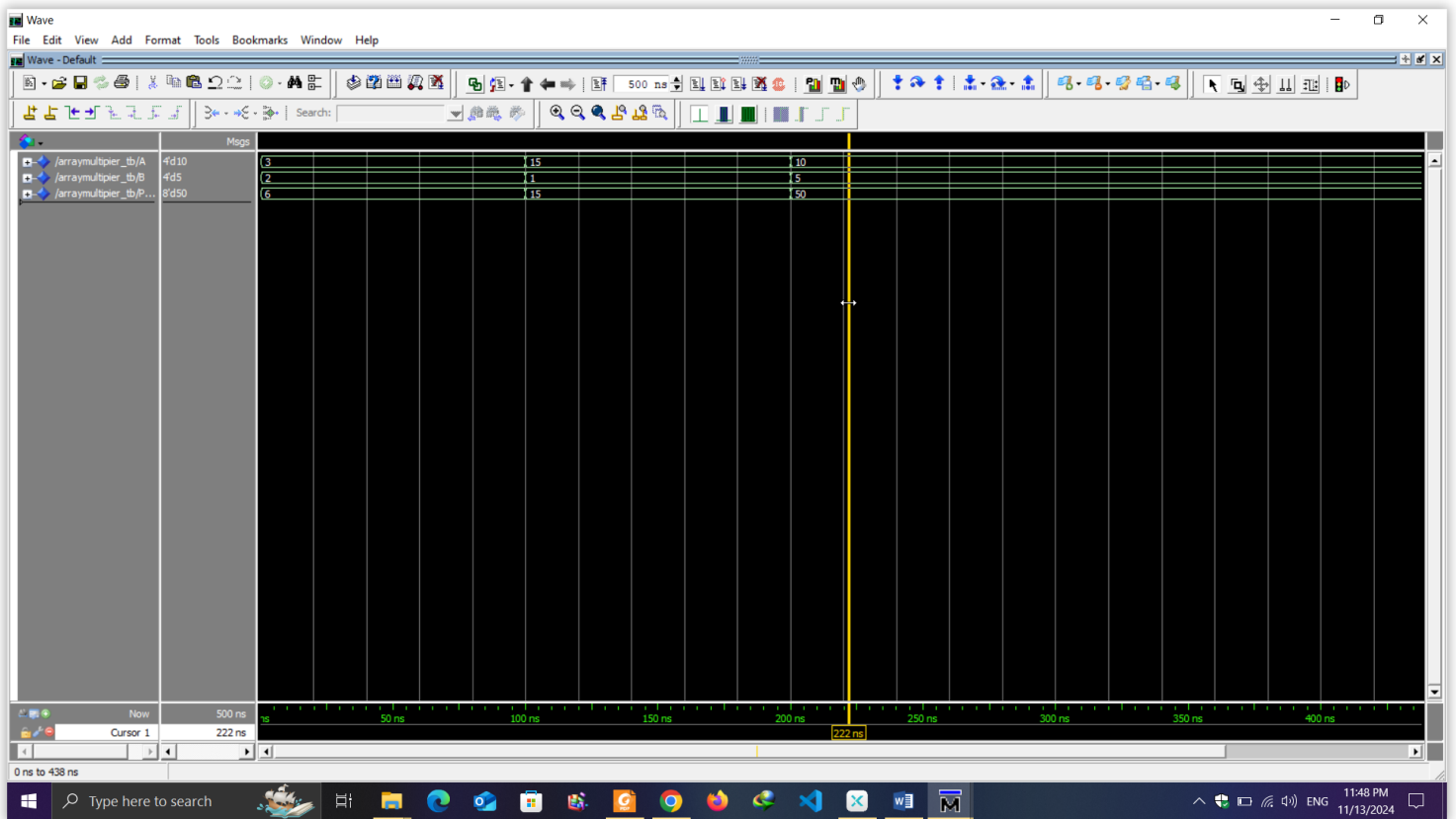
FA simulate



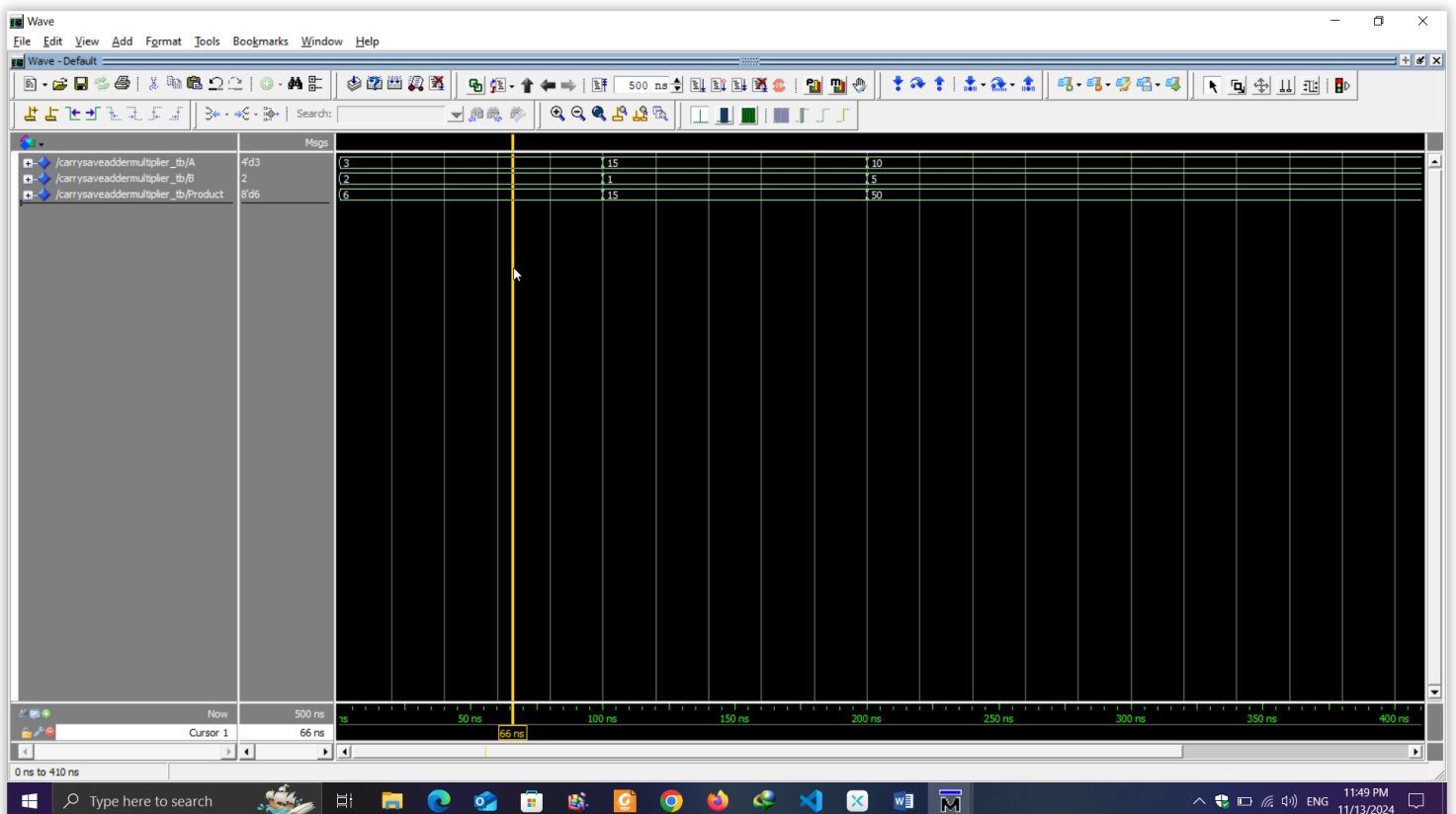
Ripple Adder simulate



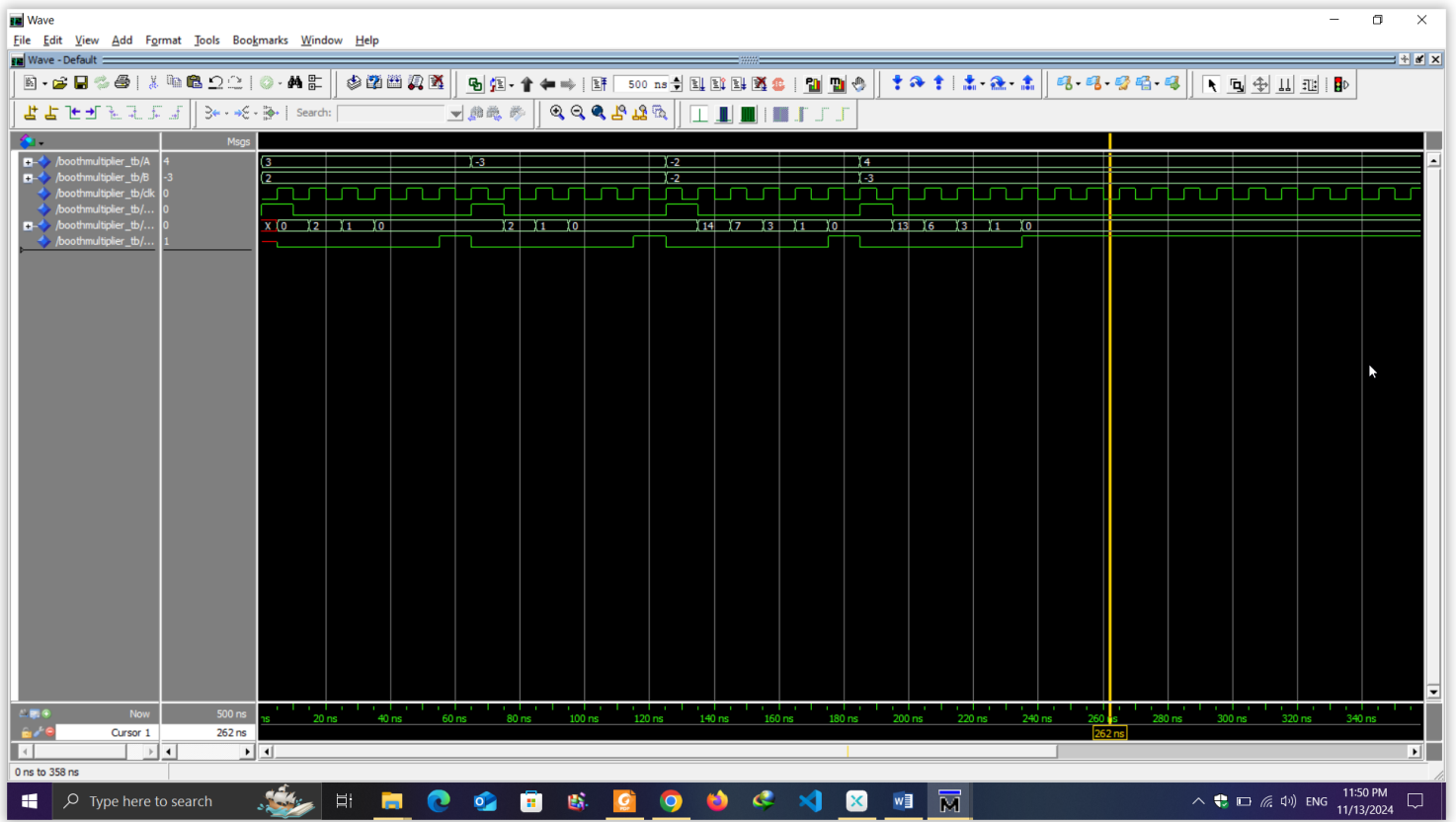
Ordinary Multiplier simulate



Array Multiplier simulate



Carry Sava Adder Multiplier simulate



Both Multiplier simulate