

در این آزمایش ابتدا ما باید واحد Half Adder را طراحی نماییم که بعد با استفاده از آن واحد Full Adder را که واحد های جمع کننده در جمع کننده ابشاری (Ripple Adder) و در جمع کننده انتخابی (Carry Select Adder) هستند را پیاده سازی کنیم.

بعد از طراحی و نوشتن کد این دو جمع کننده (HA, FA) نوبت به طراحی جمع کننده های ابشاری که با نام (Ripple) و یا با نام (Cascaded Adder) نیز شناخته میشوند؛ میرسد. این جمع کننده از اتصال چندین FA به صورت ابشاری به هم ساخته میشود که در آن carry از مرحله قبل به مرحله بعد منتقل میشود. در نهایت یک بیت carry خروجی میدهد که نشان دهنده اینست که جمع دو عدد n بیتی در n بیت جا نشده و یک بیت carry داده است.

حال نوبت به طراحی جمع کننده (Carry Look-Ahead Adder) میرسد. این جمع کننده بر خلاف دو جمع کننده دیگر از FA ها به عنوان واحد شمارشگر استفاده نمیکند و از سه مرحله تشکیل شده است:

مرحله اول: محاسبه $\text{carry-Generate} = A.B$ و محاسبه $\text{carry-Propagate} = A+B$ است به ازای هر بیت.

مرحله دوم: محاسبه Carry ها است که با استفاده از فرمول زیر به ازای هر بیت محاسبه میگردد:

$$C_i = G_i + (P_i.G_{i-1}) + (P_i.P_{i-1}.G_{i-2}) + \dots + (P_{i-1}.P_{i-2} \dots P_0.C_{in})$$

مرحله سوم: محاسبه Sum هاست که با استفاده از این فرمول ($S_i = A_i \text{ xor } B_i \text{ xor } C_{i-1}$) محاسبه میگردد.

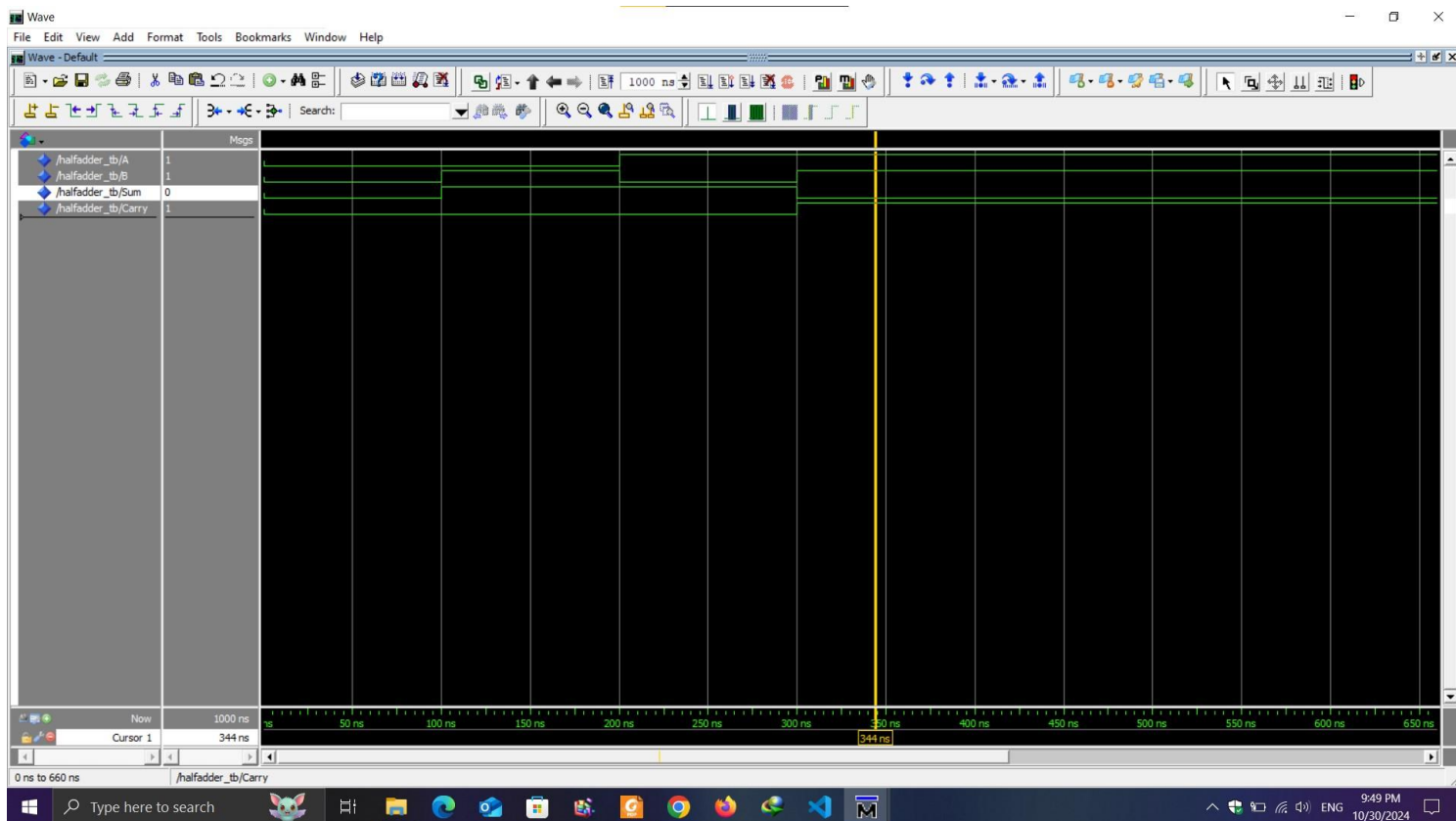
حال نوبت به طراحی آخرین جمع کننده میرسد که جمع کننده انتخابی است. این جمع کننده نیز با استفاده از واحد های جمع کننده FA ساخته میشود؛ البته انواع متفاوتی دارد که ما در آزمایش و طراحی، کد جمع کننده انتخابگر n-bit را پیاده سازی کرده ایم.

این نوع جمع کننده انتخابگر از دو تا mux که اولی بین sum ها انتخاب انجام میدهد و ورودی و خروجی آن n-bit است استفاده میکند و دومی بین carry های خروجی از هر Ripple Adder انتخاب انجام میدهد که ورودی و خروجی آن تک بیت است. ما از دو Ripple Adder به صورت موازی با هم استفاده میکنیم که اولی ورودی C_{in} آن صفر است و در دومی ورودی آن یک است؛ بعد بیت ورودی C_{in} اصلی را به خط select انتخابگر ها متصل میکنیم که بر اساس آن خروجی sum, carry ما را انتخاب کنند.

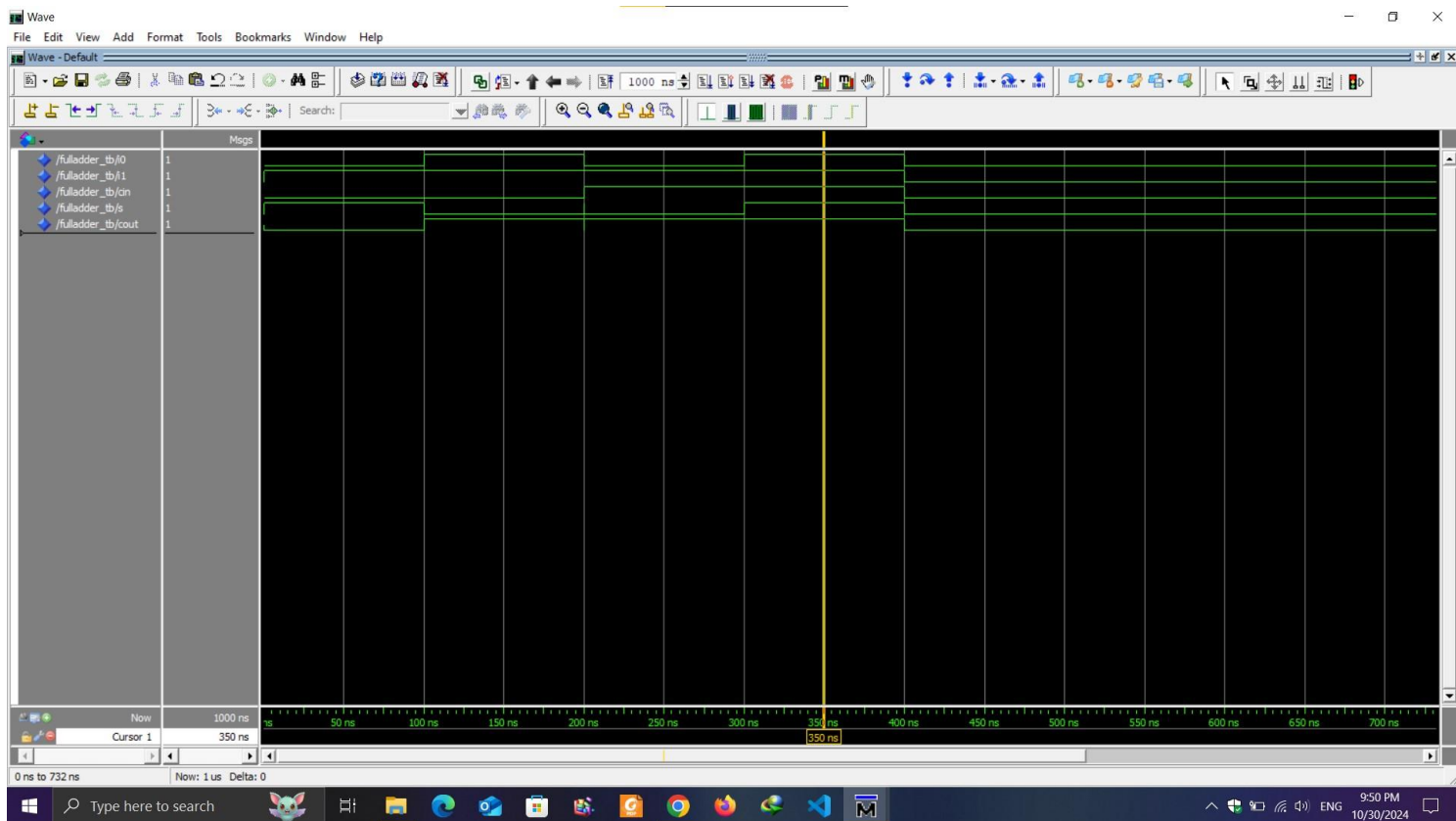
البته این در طراحی سخت افزاری آن است و ما در کدمان به جای استفاده از کد یک mux از یک process استفاده کرده و ورودی ($C_{in}, \text{sum}0, \text{sum}1, \text{cout}0, \text{cout}1$) را به عنوان خطوط حساس به آن داده ایم که براساس ساختار شرطی میاید و ورودی را به خروجی انتقال میدهد.

---جمع کننده های ما در این آزمایش 4 بیتی هستند.

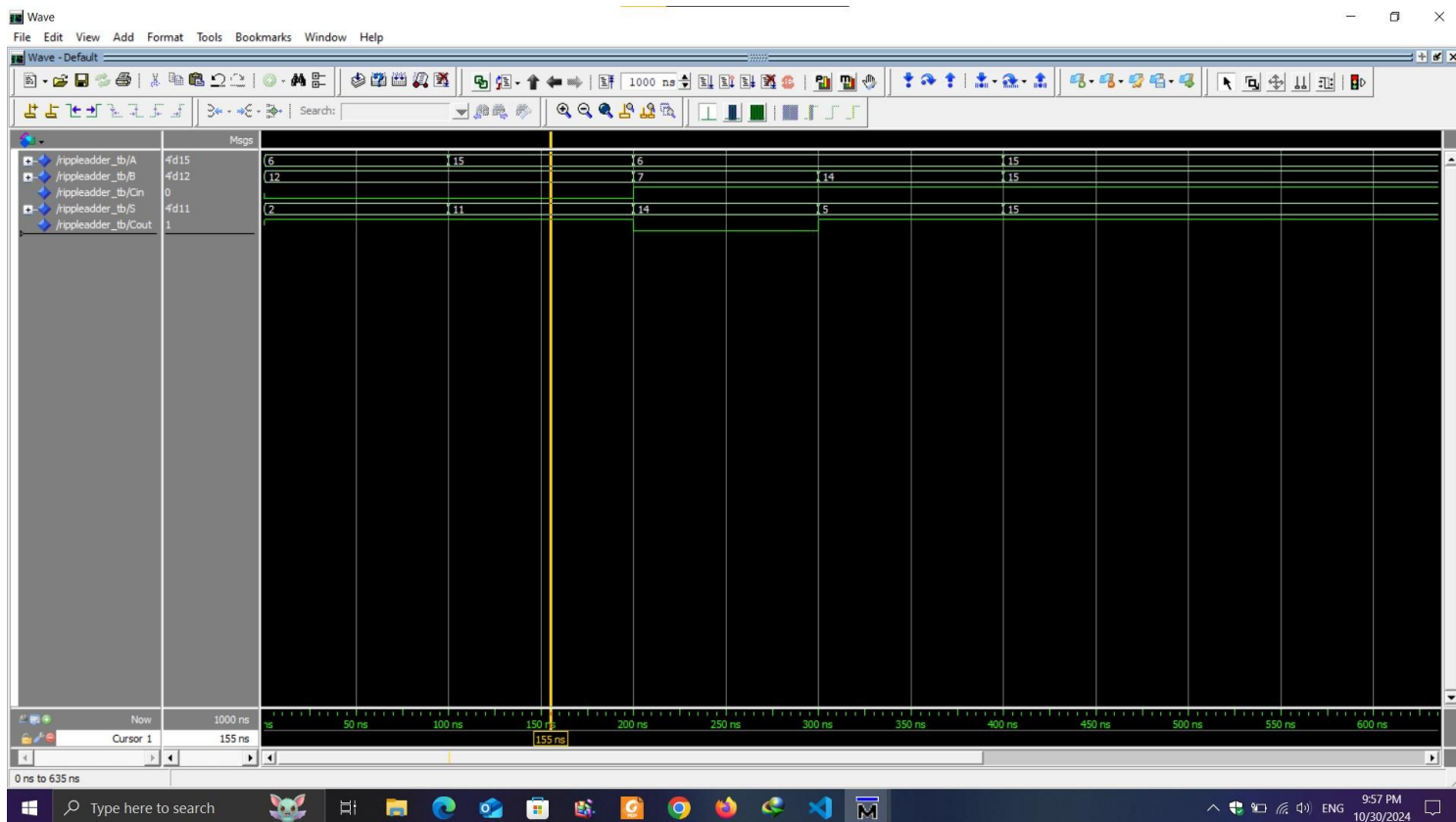
.....در ادامه عکس از نتایج شبیه سازی های هر کدام از گیت ها قرار داده شده است:



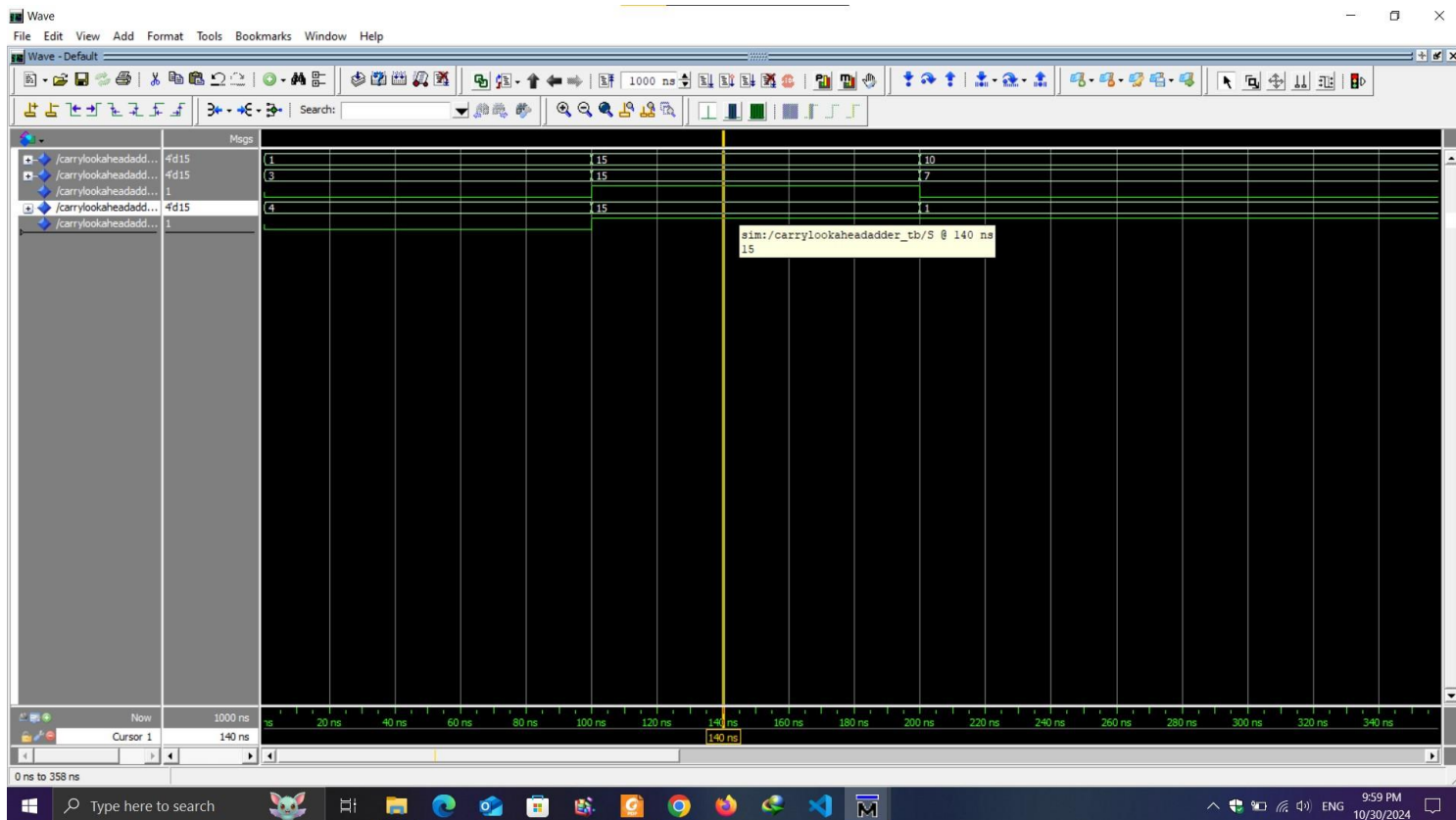
Half Adder simulate



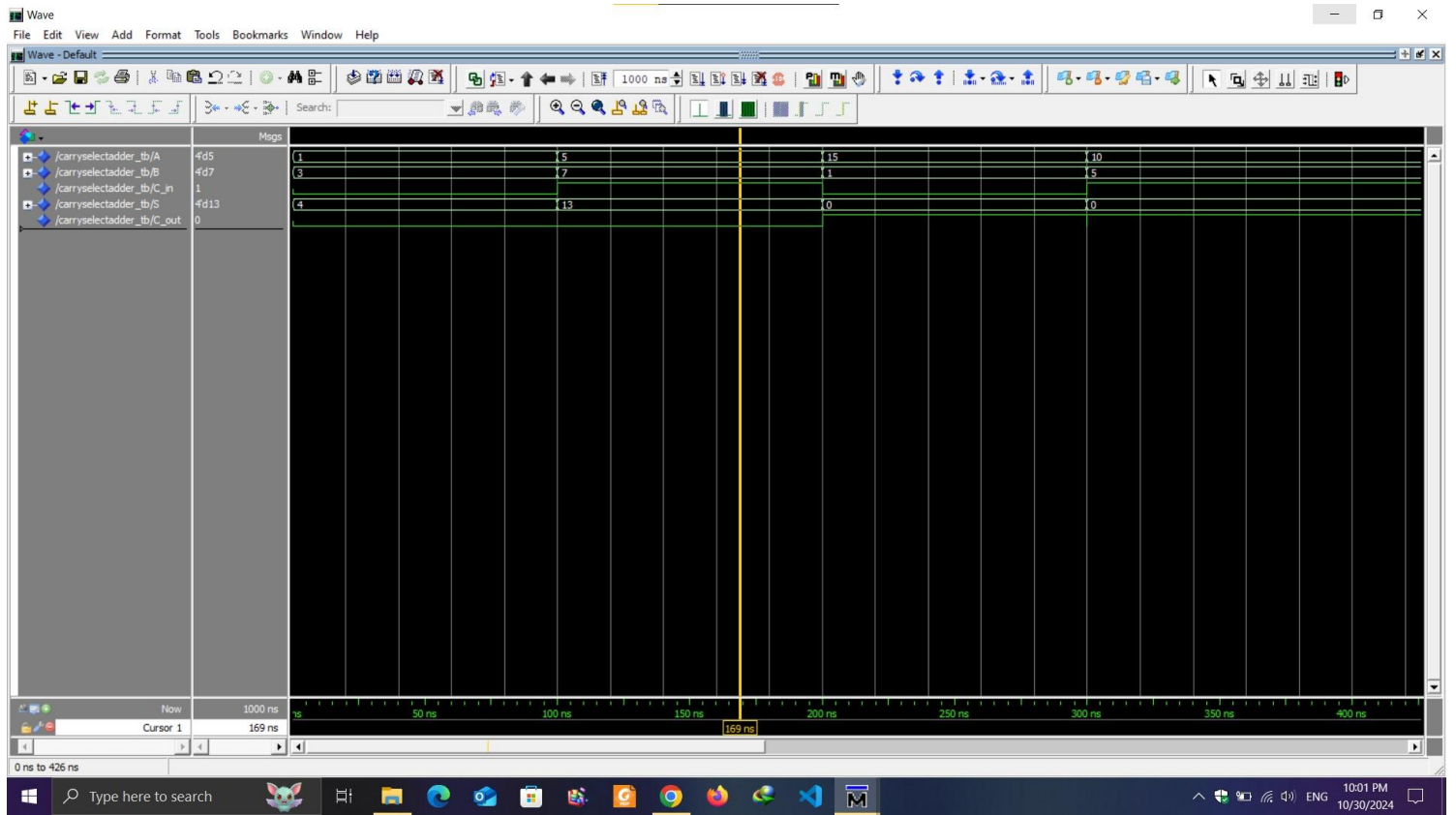
Full Adder simulate



Ripple Adder simulate



Carry Look-Ahead Adder simulate



Carry Select Adder simulate