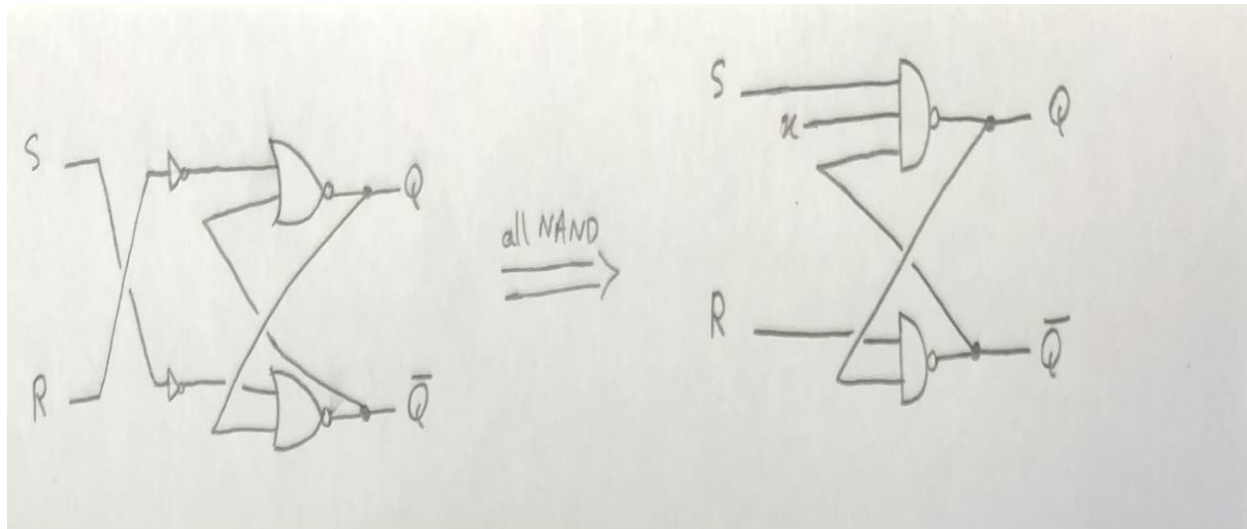


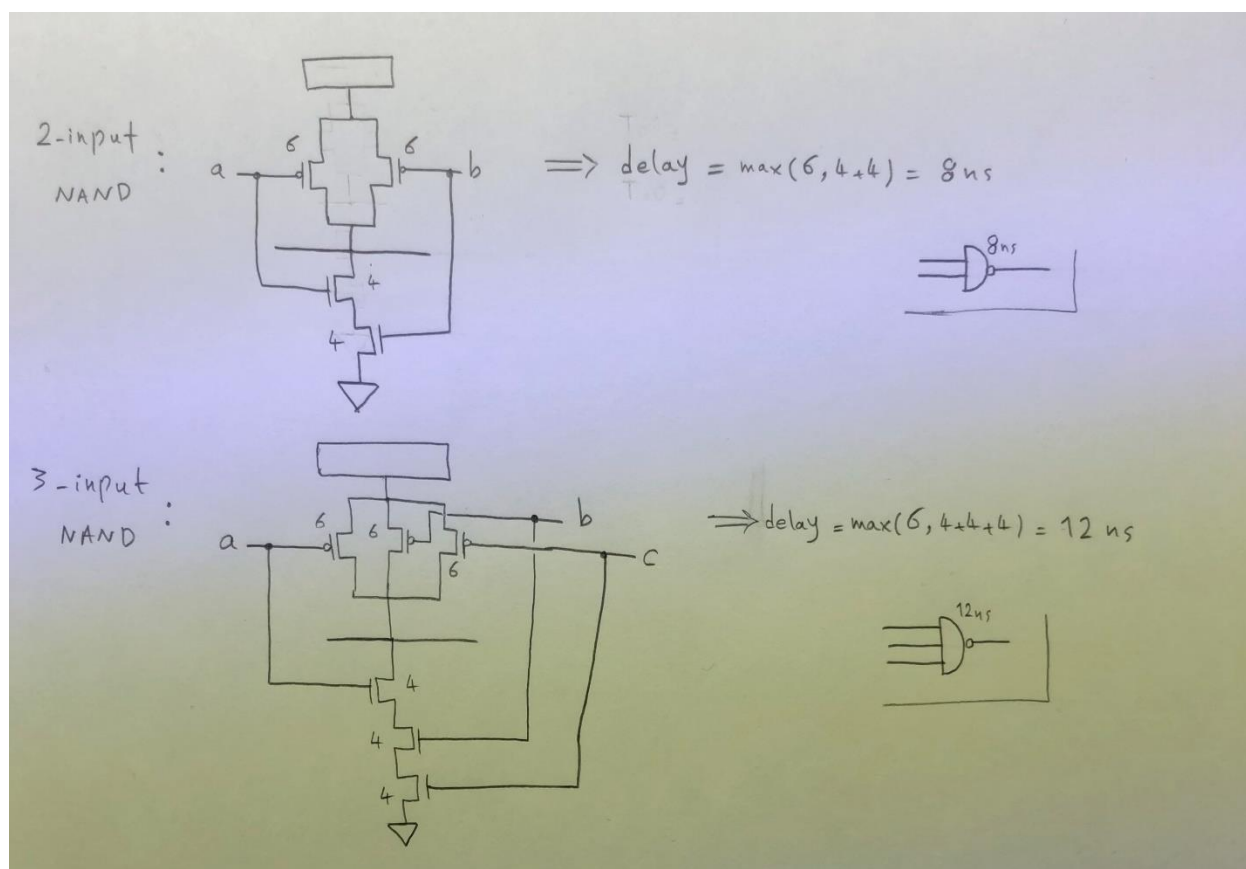
سوال 1) a. ابتدا یک SR latch معمولی با NOR در نظر میگیریم. چون ورودی ها active low هستند جلوی هر ورودی یک inverter قرار میدهیم. سپس مدار را ساده میکنیم و باجابه جایی bubble ها به مدار all NAND زیر میرسیم:



همانطور که در سوال گفته شده NAND gate بالایی را 3 ورودی میکنیم تا در صورت نیاز از آن استفاده کنیم. اگر هم نخواستیم استفاده کنیم به آن 1 میدهیم یعنی $x = 1$.

البته چون در سوال 2 فقط یک gate بالا 3 ورودی است از این ساختار اضافه میکنیم و برای سوال 3 که تمام gate ها 3 ورودی هستند تمام gate ها را 3 ورودی قرار میدهیم.

b. delay های nmos و pmos داده شده و میتوان با آن ها delay های هر دو NAND gate با 2 و 3 خروجی را با رسم switch level هر کدام از آن ها طبق شکل زیر محاسبه کرد:



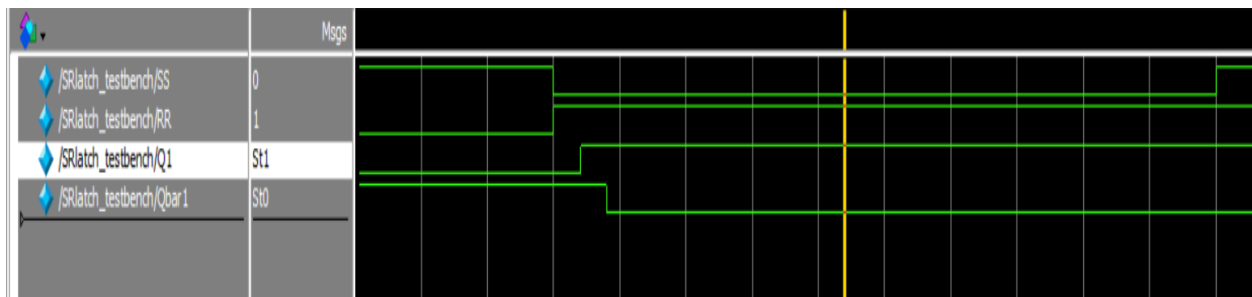
در نتیجه worst case delay این ساختار SR latch با یک gate 3 ورودی و یک 2 ورودی برابر $\max(8, 12) = 12 \text{ ns}$ و نیز با دو gate 3 ورودی برابر $\max(12, 12) = 12 \text{ ns}$ خواهد بود و لذا در هر دو حالت ما delay برابر با 12 ns خواهیم داشت.

c. در این latch در waveform ها طبق چیزی که انتظار داریم:

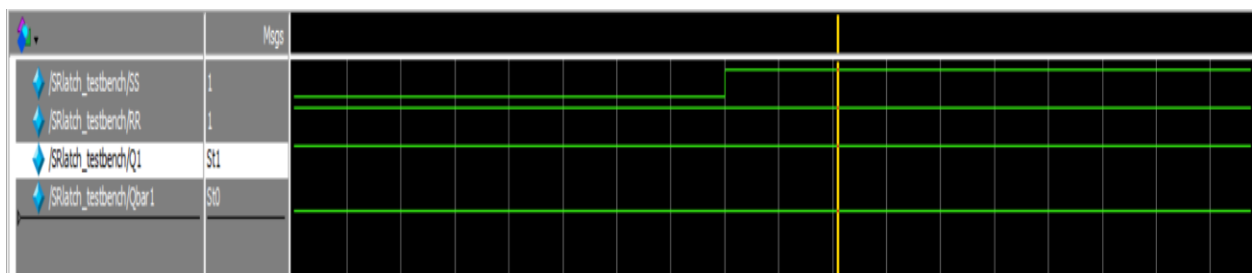
اگر $S = 1$ و $R = 0$ آنگاه reset رخ داده و $Q = 0$ میشود:



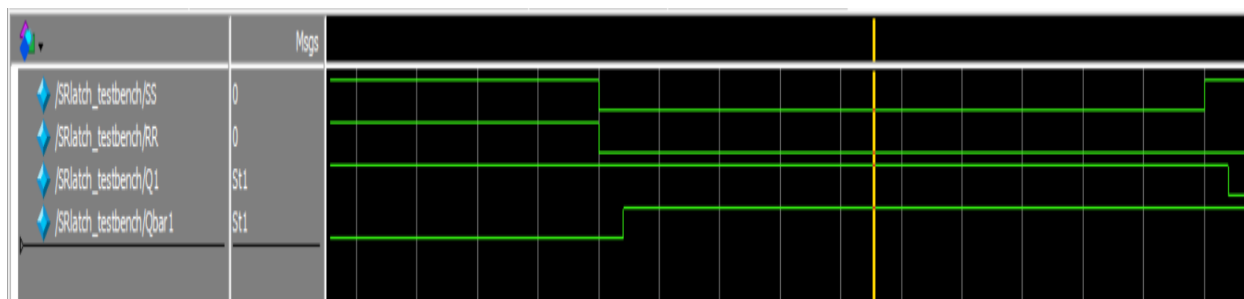
اگر $S = 0$ و $R = 1$ آنگاه set رخ داده و $Q = 1$ میشود:



اگر $S = 1$ و $R = 1$ آنگاه حالت قبلی Q حفظ میشود:



اگر $S = 0$ و $R = 0$ آنگاه حافظه مدار از بین میرود و Q و \bar{Q} هر دو برابر 1 میشوند:



در نتیجه در این حالت آخر memory loss قابل مشاهده میباشد.

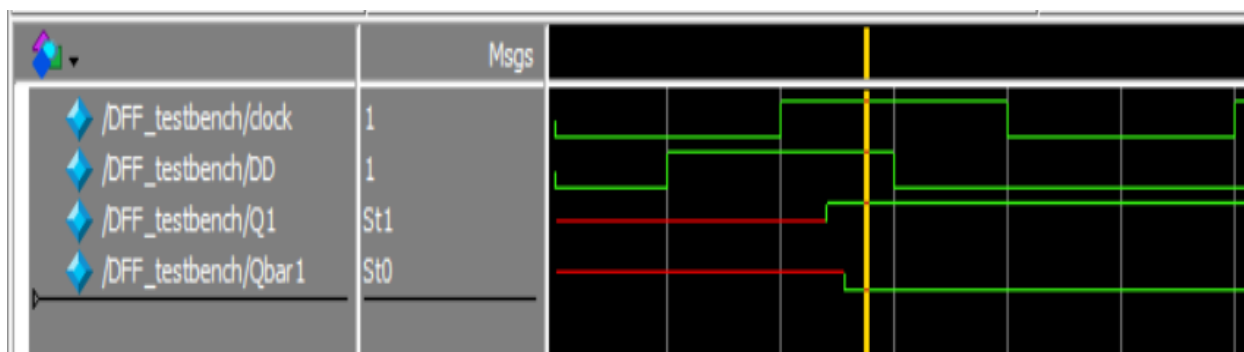
سوال 2) a. در این flip flop ما در یکی از latch ها در gate بالایی از 3 NAND ورودی استفاده کردیم. برای اینکه delay ها یکسان باشند برای هر 3 تا latch از 3 ورودی ها استفاده میکنیم و در 2 تای آن ها به جای ورودی سوم 1 را میدهم.

پس کافیه فقط 3 بار module سوال 1 را استفاده کنیم که از gate های 3 ورودی استفاده شده.

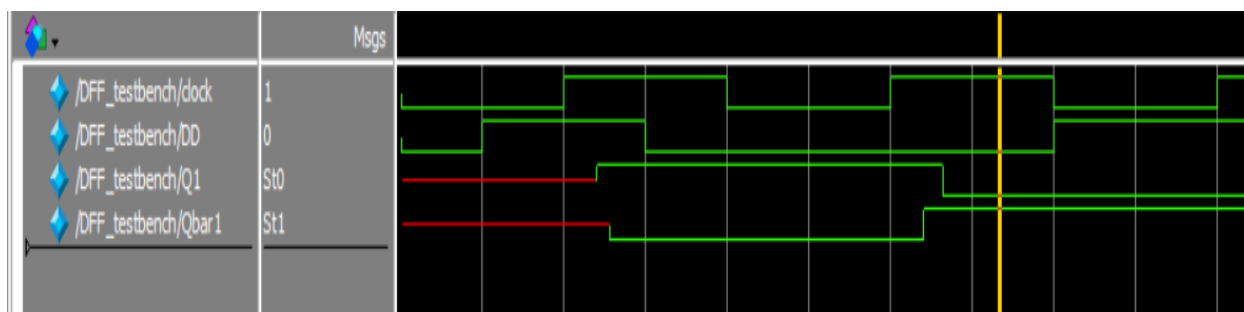
b. همانطور که در wave های زیر معلوم است clock ما rising edge است زیرا فقط هنگام تغییر clock از 0 به 1 در خروجی تغییر ایجاد میشود. پس یک D flip flop rising edge داریم.

پس از هر clock edge تغییرات Q با توجه به D به شکل زیر است:

اگر در حوالی تغییر clock از 0 به 1 مقدار $D = 1$ باشد آنگاه $Q = 1$ می شود:

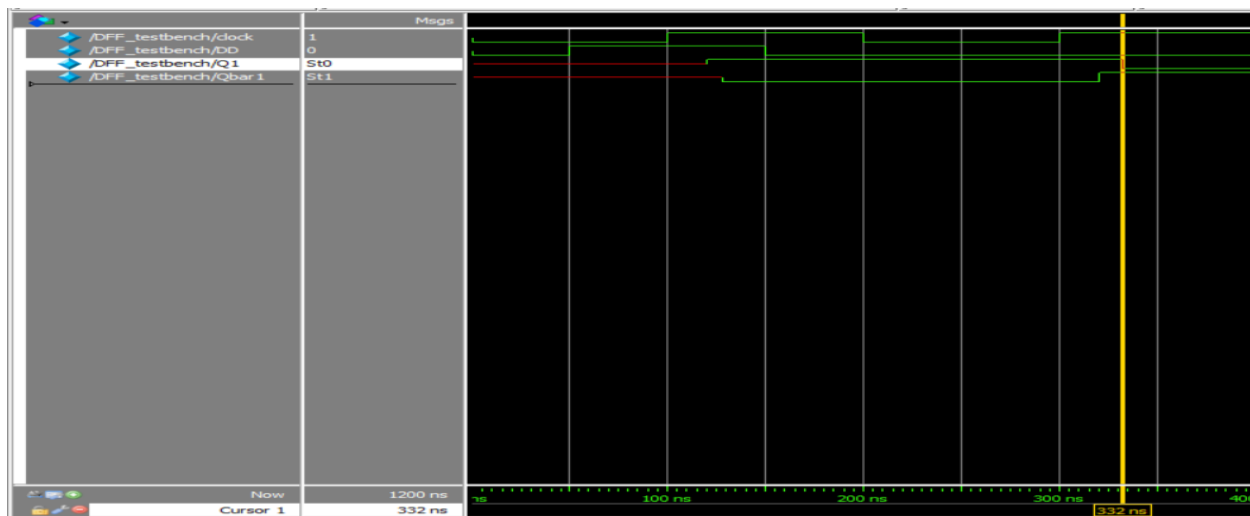
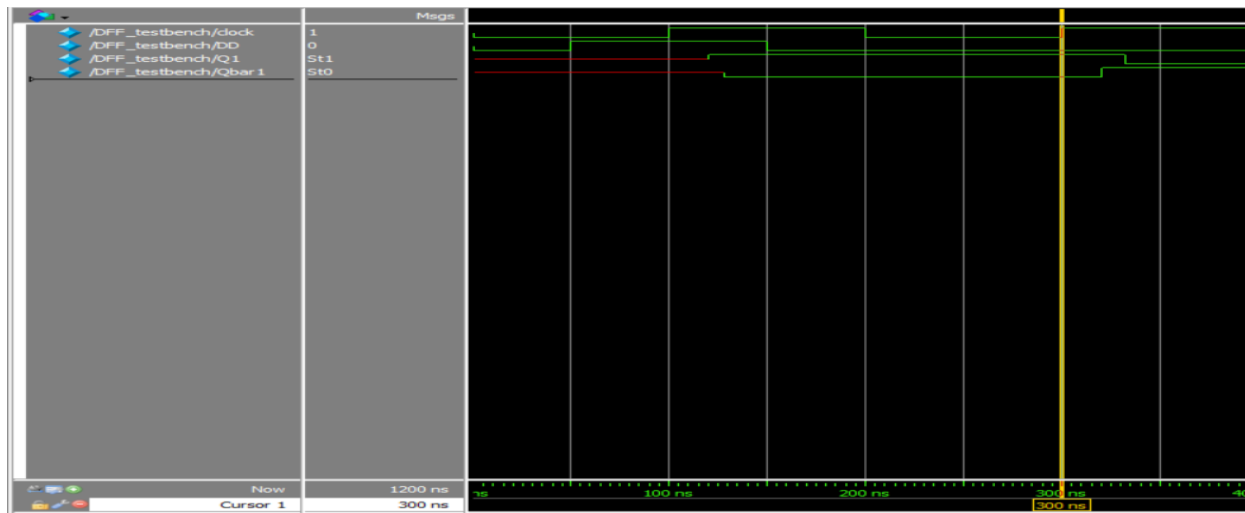


اگر در حوالی تغییر clock از 0 به 1 مقدار $D = 0$ باشد آنگاه $Q = 0$ می شود:

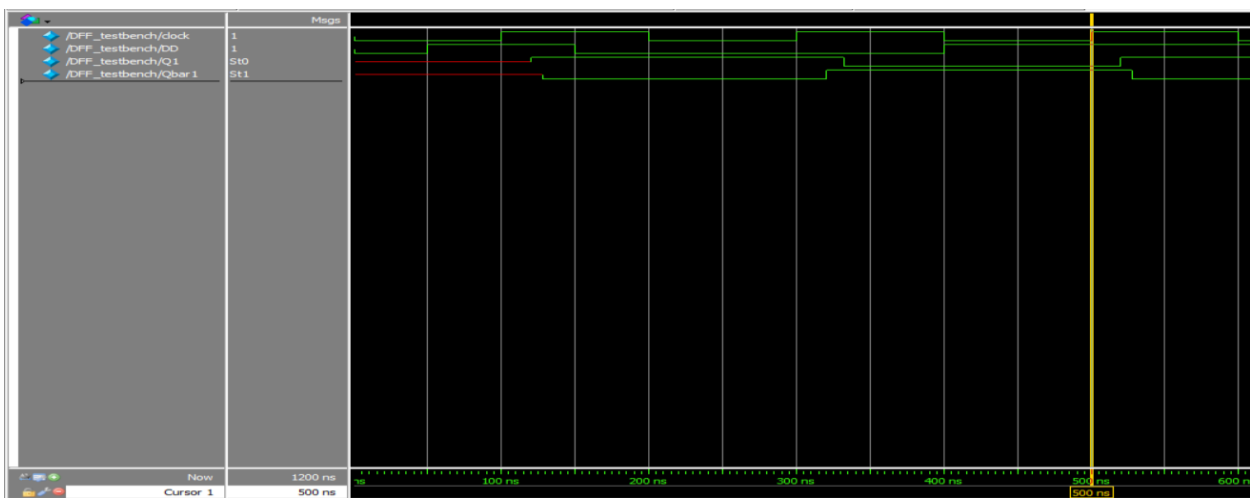


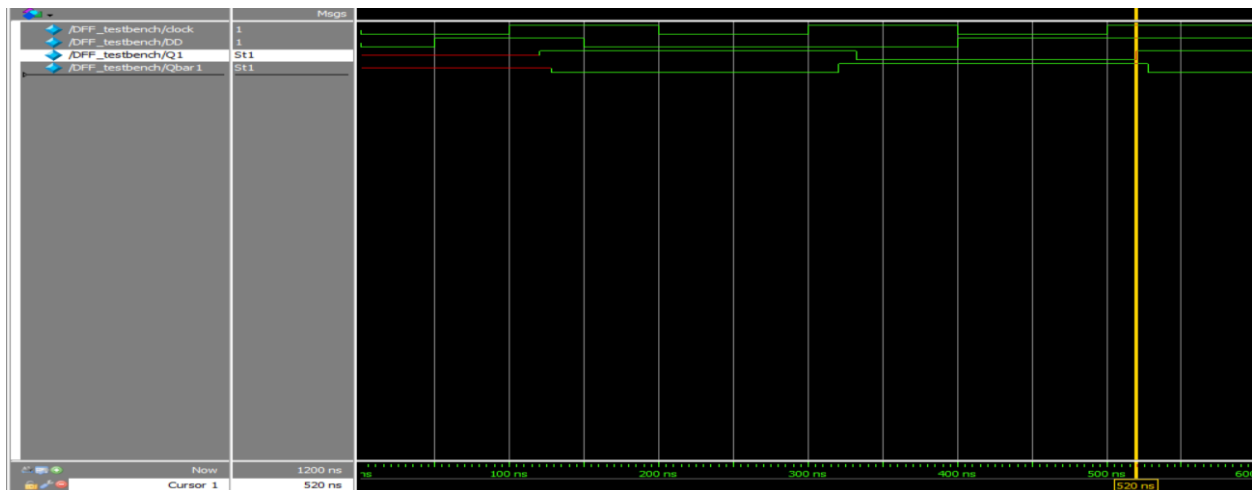
هنگام تغییر clock خروجی Q با delay تغییر میکند. $TO0$ و $TO1$ آن به شکل های زیر محاسبه میشود:

T01 برابر $332 - 300 = 32\text{ns}$ می باشد:

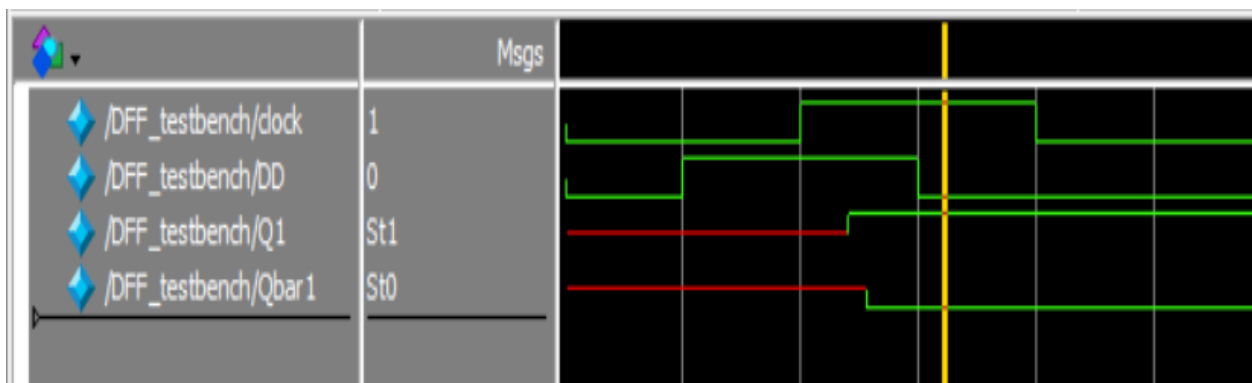


T00 برابر $520 - 500 = 20\text{ns}$ می باشد:





همانطور که مشخص است هنگام active بودن clock با تغییر دادن مقدار D هیچ تغییری در خروجی Q ایجاد نمیشود.
مثلا در این wave در میانه ی clock مقدار D از 1 به 0 تغییر کرده اما مقدار Q همچنان 1 باقی مانده است:



c. طبق تعریف t setup یعنی حداقل زمانی که باید میان تغییر clock و D فاصله باشد. طبق اندازه گیری ها این اندازه حداقل باید $12 + 1 = 13\text{ns}$ باشد تا clock بتواند مقدار جدید D را روی Q قرار بدهد وگرنه Q مقدار قبل ترش را حفظ میکند.

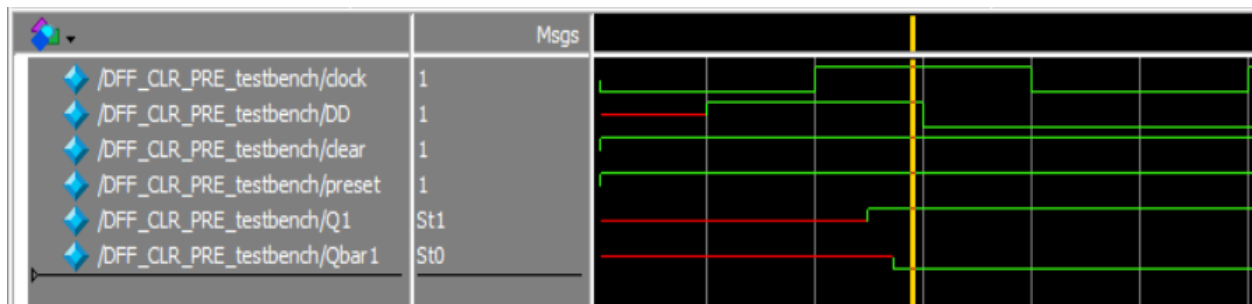
d. در اینجا نیز $t_{hold up}$ که یعنی حداقل زمانی که باید میان تغییر clock و بعد از D فاصله باشد تا D درست روی Q برود. این مقدار نیز برابر $0 + 1 = 1ns$ می باشد و حداقل $1ns$ باید فاصله باشد وگرنه اگر clock و D همزمان باهم تغییر کنند تغییر آن روی Q اعمال نمیشود.

سوال (3) e. در این flip flop تمام NAND gate ها 3 ورودی هستند پس از module های 3 ورودی استفاده میکنیم. 3 تا latch داریم یعنی باید 3 بار از آن استفاده کنیم.

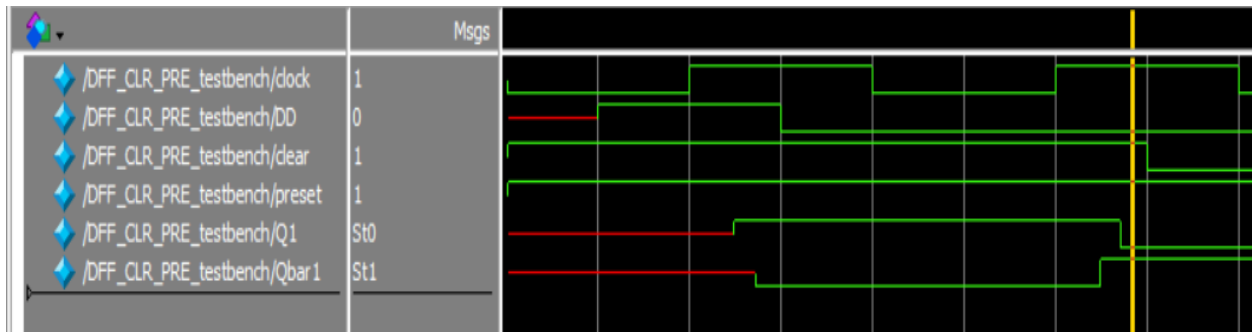
توجه شود که مقادیر clear و preset هر دو active low هستند یعنی هرکدامشان با 0 شدن active میشوند.

f. تغییرات Q با توجه به تغییر ورودی ها به شکل زیر است:

اگر $D = 1$ باشد $Q = 1$ میشود با delay برابر با $TO1 = 24ns$:

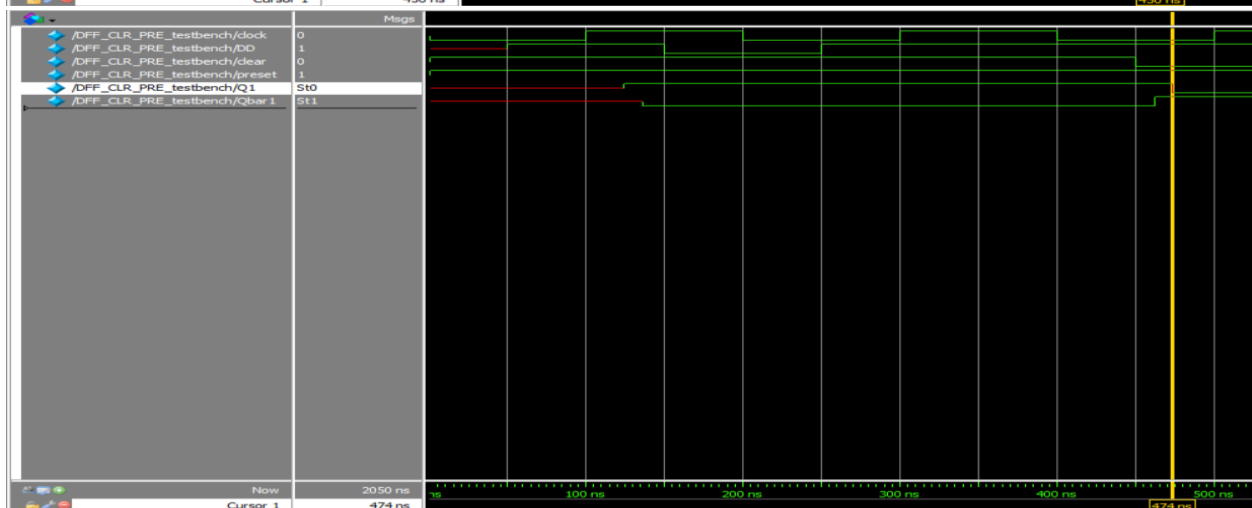
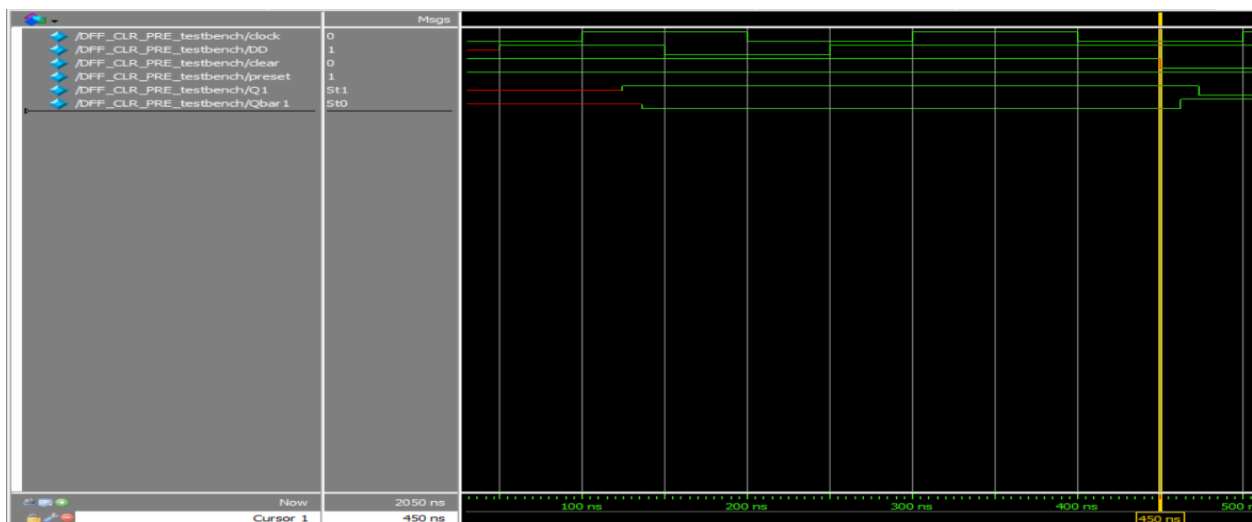


اگر $D = 0$ باشد $Q = 0$ میشود با delay برابر با $TO0 = 36ns$:



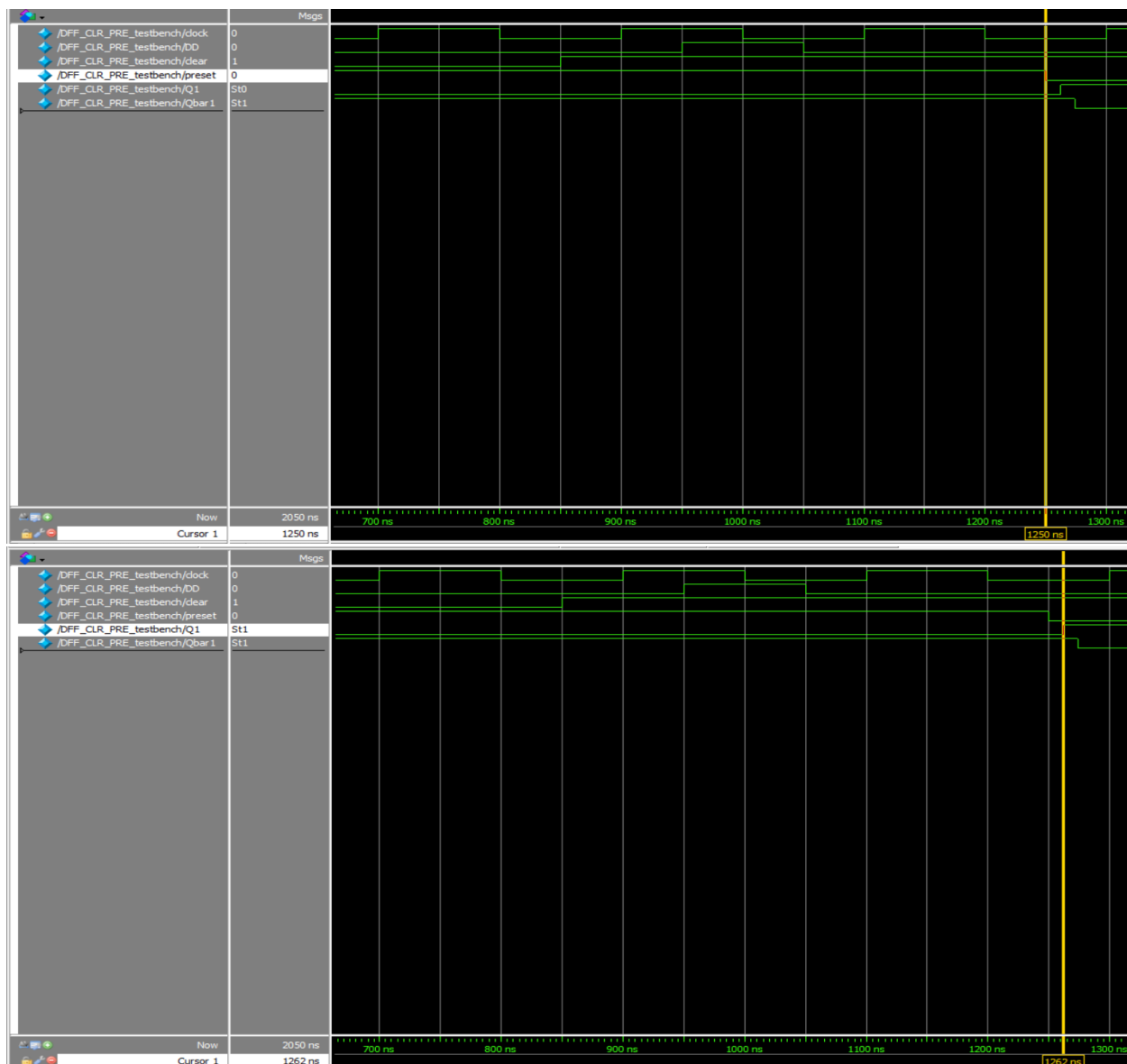
اگر $clear = 0$ یعنی active باشد آنگاه بدون توجه به clock و D مقدار $Q = 0$ میشود.

delay آن نیز برابر $450 - 474 = 24ns$ خواهد بود:



همچنین اگر $\text{preset} = 0$ یعنی active باشد آنگاه بدون توجه به clock و D مقدار $Q = 1$ میشود.

delay آن نیز برابر $1262 - 1250 = 12$ خواهد بود:



g. هنگام active بودن مثلاً preset با هر تغییر در clock و D مقدار Q تغییری نمیکند و همان 1 باقی میماند:



h. با همزمان active بودن هر دو clear و preset مدار حافظه خود را از دست میدهد و هر دو Q و Qbar مقدار 1 را میگیرند و لذا loss of memory رخ خواهد داد:

