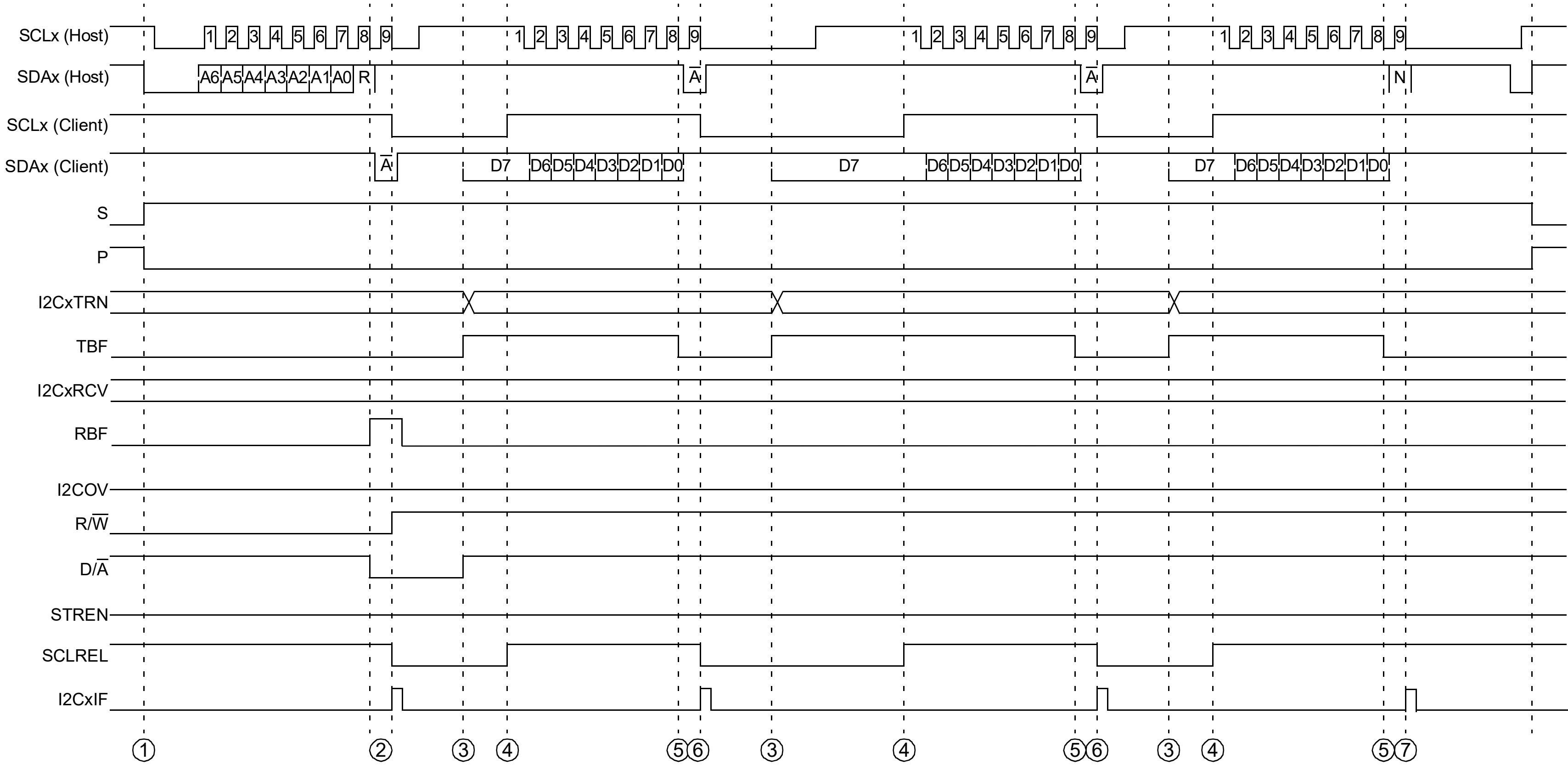

① Client recognizes Start event, S and P bits set/clear accordingly.

② Client receives address byte. Address matches. Client generates interrupt. Address byte is moved to I2CxRCV register and is read by user software to prevent buffer overflow. R/$\overline{W}$ = 1 to indicate read from client. SCLREL = 0 to suspend host clock.

③ User software writes I2CxTRN with response data. TBF = 1 indicates that buffer is full. Writing I2CxTRN sets D/$\overline{A}$, indicating a data byte.

④ User software sets SCLREL to release clock hold. Host resumes clocking and client transmits data byte.

⑤ After last bit, module clears TBF bit, indicating buffer is available for next byte.

⑥ At the end of ninth clock, if the host has sent an $\overline{ACK}$, module clears SCLREL to suspend clock. Client generates interrupt.

⑦ At the end of ninth clock, if host sent a NACK, no more data expected. User software should stop writing to I2CxTRN. Module does not suspend clock and will generate an interrupt.