

α), β) Δες αρχείο .sv

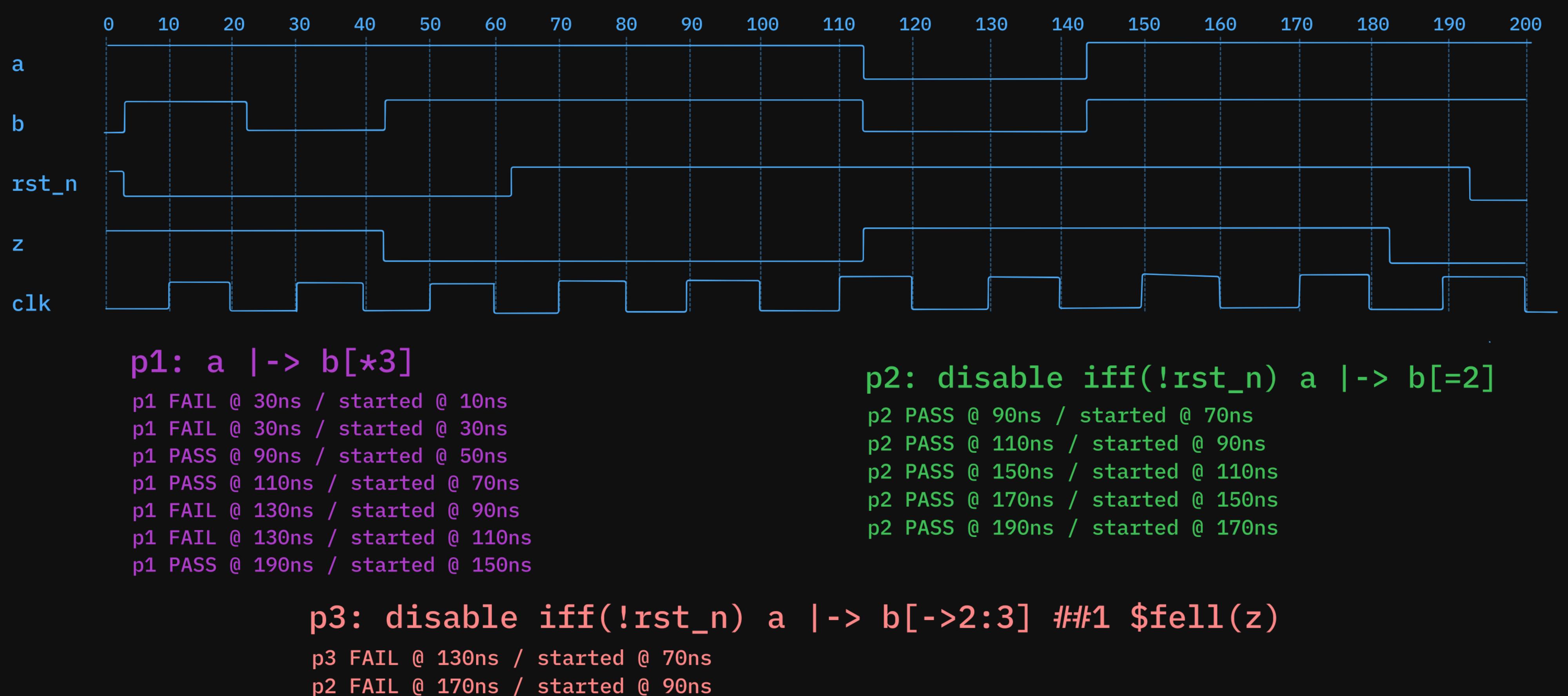
γ) Στην συγκεκριμένη άσκηση έχουμε clock domain crossing, δηλαδή μεταφέρουμε δεδομένα από μία περιοχή ρολογιού (clk1) σε μια άλλη (clk2).

Το πρόβλημα εδώ είναι ότι το q1 μπορεί να αλλάξει τιμή σε μια χρονική στιγμή που το clk2 μεταβαίνει στο posedge. Στην ουσία έχουμε time violation και μεταβαίνουμε σε μια κατάσταση "μετασταθερότητας" που το σήμα q2 δεν είναι ούτε 0 ούτε 1.

Με το DFF3 (με έξοδο q3) δίνουμε χρόνο στο κύκλωμα να επιλύσει αυτήν την αβεβαιότητα, αλλά πάλι υπάρχει πιθανότητα να μην έχει επιλυθεί μέχρι την χρονική στιγμή που το q3 (δηλ το Enable) εισέρχεται στο mux. Αν το q3 βρίσκεται ακόμα σε metastable state, τότε δεν ξέρουμε τι θα γίνει στο mux.

Για να επιλυθεί αυτό το πρόβλημα, η καλύτερη λύση είναι να δημιουργήσουμε μια *Asynchronous FIFO* με write clock το clk1, και read clock το clk2. Έτσι μπορούμε να μεταφέρουμε δεδομένα από το ένα clock domain στο άλλο με ασφάλεια.

Εναλλακτικά, μπορούμε να προσθέσουμε παραπάνω FF μετά το q3 για να του δώσουμε παραπάνω χρόνο, αλλά δεν θεωρείται ιδανική λύση (απλά αυξάνουμε το MTBF, δεν θεωρείται robust solution).



p2 PASS @ 190ns / started @ 110ns