ECE333: Εργαστήριο Ψηφιακών Συστημάτων

Lab 4 – ADXL362 Accelerometer Driver

Νίκας Ιάσων

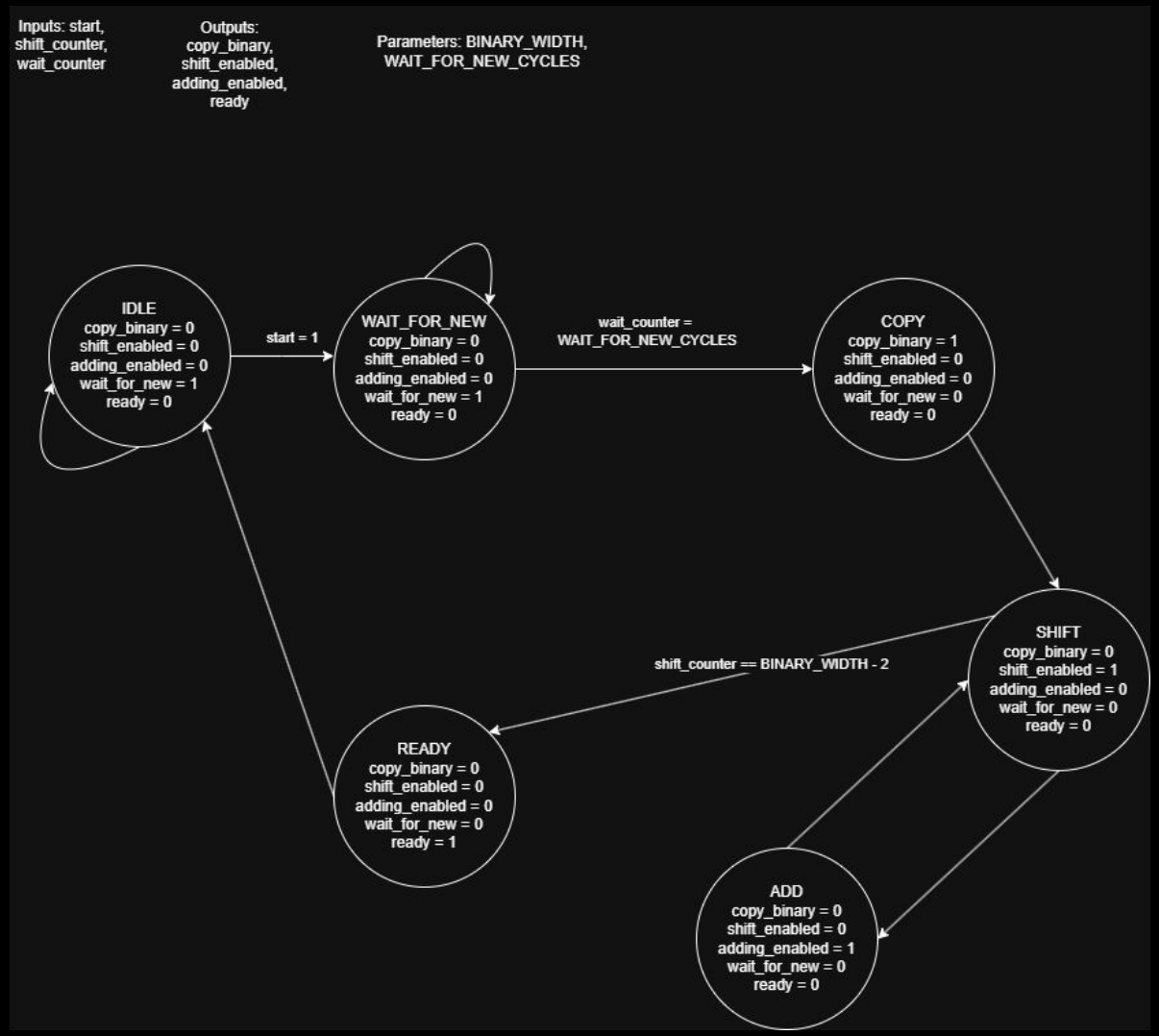
Μέρος Α – Binary to ASCII converter

Μέρος Α – Περιγραφή

Για την υλοποίηση του μετατροπέα χρησιμοποιήθηκε μια FSM η οποία δέχεται 1 τιμή σε binary (12 και 19 bits), βρίσκει το πρόσημο του και κάνει μετατροπή της απόλυτης τιμής του αριθμού σε BCD κωδικοποίηση (4 και 6 ψηφία) χρησιμοποιώντας τον αλγόριθμο Double dabble. Για να ολοκληρωθεί ο αλγόριθμος χρειάζεται 2(n-1) κύκλους, όπου n τα bits του αρχικού binary αριθμού.

Έπειτα τα ψηφία σε BCD μετατρέπονται σε τιμές ASCII προσθέτοντας την τιμή 48 που είναι η τιμή του χαρακτήρα ‘0’ στο σύστημα ASCII. Μόλις αυτές οι τιμές είναι έτοιμες ένα σήμα ready σηκώνεται ώστε να ξεκινήσει η μετάδοση των χαρακτήρων απο το uart_transmitter module.

Μέρος Α – FSM diagram



Για την FSM χρησιμοποιήθηκαν 6 states. Αρχικά το IDLE state στο οποίο h FSM περιμένει το σήμα start να γίνει 1 για να ξεκινήσει μια μετατροπή.
Έπειτα πηγαίνει στο WAIT_FOR_NEW state στο οποίο παραμένει μέχρι να έχουν περάσει 5 κύκλοι από την ολοκλήρωσει της τελευταίας μετατροπής ώστε να είναι σίγουρο ότι ο uart_transmitter έχει διαβάσει σωστά τις προηγούμενες τιμές. Για έναν κύκλο παραμένει στο state COPY όπου αντιγράφονται οι τιμή του καινούριου binary και γινονται reset οι BCD registers. Έπειτα εναλλάξ για n-1 φορές, όπου n τα bits του αρχικού binary αλλάζει μεταξύ των states SHIFT και ADD. Τέλος, πηγαίνει για έναν κύκλο στο READY state.

Μέρος Α – Simulation και προβλήματα

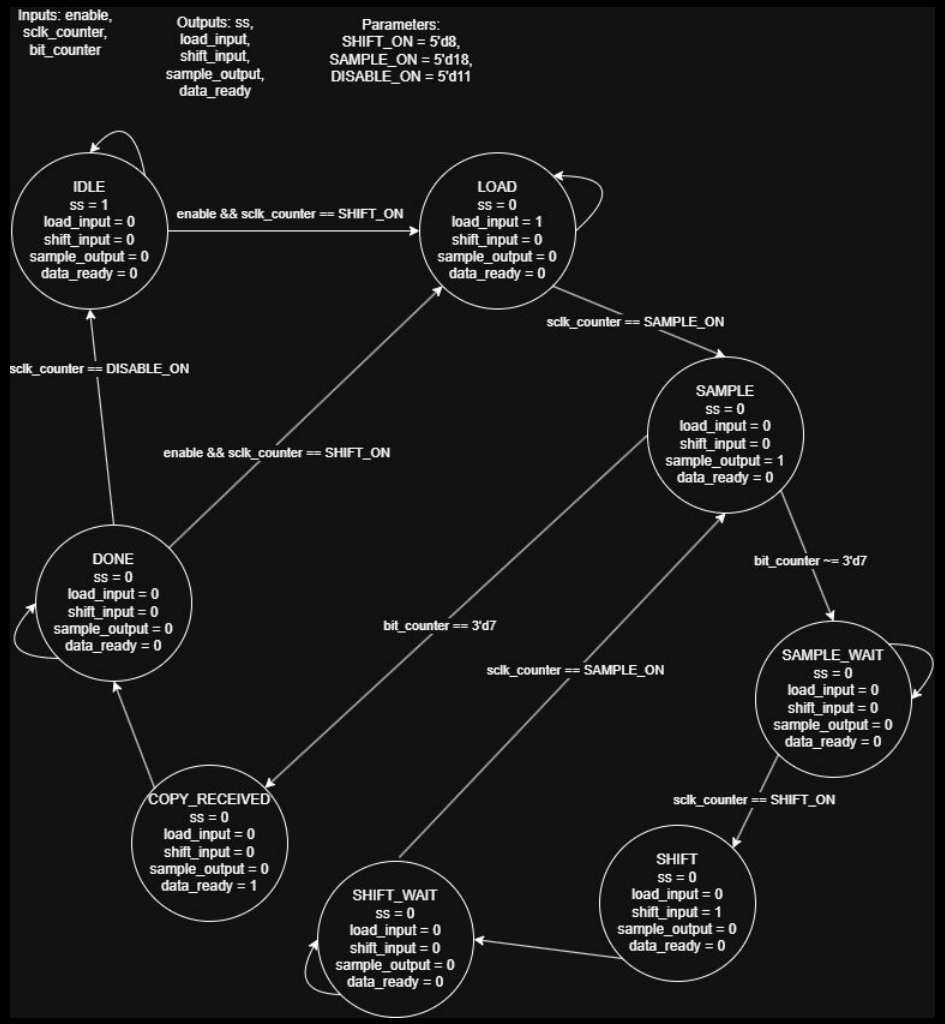
Για την προσομοίωση χρησιμοποιήθηκε ένα testbench το οποίο δίνει τιμές στον converter και ελέγχει το σήμα start που έχει ως input ο converter. Έτσι δοκιμάστηκαν πολλές διαφορετικές τιμές και εντοπίστηκαν 2 προβλήματα με το αρχικό design.
Το πρώτο αφορά τις διαδοχικές μετατροπές αριθμών σε BCD και παρατηρήθηκε πως τα BCD registers δεν γίνονταν 0 για κάθε καινούρια μετατροπή με αποτέλεσμα οι επόμενες μετατροπές να μην βγάζουν σωστό αποτέλεσμα.
Το δεύτερο αφορούσε της μεταβάσεις τις FSM και ήταν ένα off-by-one error γιατί το ADD state πραγματοποιούνταν n+1 φορές με αποτέλεσμα να στα τελικά ψηφία που είναι μεγαλύτερα του 4, να προστίθεται μια παραπάνω φορά το 3.

Μέρος Β – SPI Master

Μέρος Β – Περιγραφή

Για την δημιουργία του ρολογιού sclk που απαιτείται για την επικοινωνία, χρησιμοποιήθηκε το MMCM module που υπάρχει στην πλακέτα και με τις κατάλληλες ρυθμίσεις δημιουργήθηκε ένα ρολόι 5MHz. Επιπλέον χρησιμοποιήθηκε ένας μετρητής που λειτουργεί με το γρήγορο ρολόι της πλακέτας και ξεκινάει όταν το αργό ρολόι είναι σταθερό. Αυτός ειναι που μας δίνει την δυνατότητα να ξέρουμε που βρίσκεται το αργό ρολόι ώστε να μπορούμε να συχγρονίσουμε τα σήματα του προτοκόλλου σωστά. Για τον συντονισμό αυτών των σημάτων χρησιμοποιήθηκε μια FSM η οποία συντονίζει πότε ξεκινάει μια νέα αποστολή/λήψη, πότε γίνεται η ανάγνωση του σήματος MISO, πότε ανανεώνεται η τιμή του σήματος MOSI και σύμφωνα με ένα σήμα εισόδου καθορίζει πότε πρέπει να τελειώσει η επικοινωνία.

Μέρος Β – FSM diagram



Μέρος Γ – ADXL362 Controller

Τελικό αποτέλεσμα