

Αρχιτεκτονική Υπολογιστών

Αναφορά Πρώτου Μέρους του Project

Ομάδα Εργασίας: Βαϊλάκης Αποστόλης & Φωτάκης Τζανής

Διεπαφές Μονάδων

Reservation Station

| Σήμα | Πλάτος (bits) | Είδος | Περιγραφή |
|----------|---------------|-------|--------------------------|
| WrEn | 1 | in | Write Enable |
| Op | 2 | in | Operation Input |
| Vj | 32 | in | Vj Input |
| Vk | 32 | in | Vk Input |
| Qj | 4 | in | Qj Input |
| Qk | 4 | in | Qk Input |
| Ex | 1 | in | Instruction was Executed |
| OpOut | 2 | out | Operation Output |
| VjOut | 32 | out | Vj Output |
| VkOut | 32 | out | Vk Output |
| ReadyOut | 1 | out | Ready to be executed |
| CDBQ | 5 | in | CDB.Q |
| CDBV | 32 | in | CDB.V |
| BusyOut | 1 | out | Reservation Station Busy |
| Rst | 1 | in | Reset |
| Clk | 1 | in | Clock |

Τεκμηρίωση Λειτουργικότητας

Το Reservation Station (RS) αποτελείται από registers οι οποίοι γράφονται μόνο όταν δεν είναι Busy κατά την έκδοση μίας εντολής προς αυτούς ή μέσω του CDB. Ένα RS θεωρείται busy όταν του έχει γραφτεί κάποια εντολή και δεν έχει λάβει ακόμα το σήμα executed από το αντίστοιχο Functional Unit. Επίσης θεωρείται Ready to be executed όταν είναι busy και οι τιμές των Qj, Qk είναι μηδενικά.

Arithmetic/Logical Functional Unit

| Σήμα | Πλάτος (bits) | Είδος | Περιγραφή |
|------------|---------------|-------|----------------|
| Clk | 1 | in | Clock |
| En | 1 | in | Enable |
| Rst | 1 | in | Reset |
| Grant | 1 | in | Grant |
| Vj | 332 | in | Vj Input |
| Vk | 32 | in | Vk Input |
| Op | 2 | in | Operation |
| Tag | 5 | in | Tag Input |
| RequestOut | 1 | out | Request to CDB |
| BusyOut | 1 | out | Busy if full |
| ResultOut | 32 | out | Result Output |
| TagOut | 5 | out | Tag Output |

Τεκμηρίωση Λειτουργικότητας

Ένα functional unit αποτελείται από την μονάδα η οποία κάνει την ζητούμενη μαθηματική πράξη και από registers που χρησιμοποιούνται σε σειρά ως καθυστερητές, αυτό συμβαίνει για την εξομοίωση ενός pipelined συστήματος. Ένα functional unit θεωρείται busy όταν κάθε register έχει tag διάφορο των μηδενικών (NOP), δηλαδή εκτελεσμένη εντολή με δεδομένα προς εγγραφή στον επόμενο ή το CDB. Το Request γίνεται μόλις εντοπιστεί κάποιο Tag διάφορο των μηδενικών στον προτελευταίο register του functional unit. Τέλος αξίζει να σημειωθεί ότι το σύστημα pipeline των functional units είναι σχεδιασμένο ώστε να συγχωνεύει τις εντολές NOP ,οι οποίες και εισέρχονται όταν το pipeline πρέπει να προχωρήσει άλλα δεν υπάρχει καινούρια εντολή για να εισαχθεί.

Arithmetic/Logical Functional Units with Reservation Station and Control

| Σήμα | Πλάτος(bits) | Είδος | Περιγραφή |
|------------|--------------|-------|----------------------------------|
| Rst | 1 | in | Reset |
| Clk | 1 | in | Clock |
| Issue | 1 | in | There are available instructions |
| Op | 2 | in | Operation Input |
| Vj | 32 | in | Vj Input |
| Vk | 32 | in | Vk Input |
| Qj | 5 | in | Qj Input |
| Qk | 5 | in | Qk Input |
| CDBQ | 5 | in | CDB.Q Input |
| CDBV | 32 | in | CDB.V Input |
| Grant | 1 | in | Grant to Write data to CDB |
| Available | 3 | out | Available to get Instructions |
| Vout | 32 | out | V to CDB |
| Qout | 5 | out | Q to CDB |
| RequestOut | 1 | out | Request to CDB |

Τεκμηρίωση Λειτουργικότητας

Μια συνολική μονάδα εντολών δέχεται από την μονάδα issue κατάλληλα πεδία Vj,Vk,Qj,Qk,Op και με την σειρά της τα βάζει σε ένα εσωρετικό Reservation Station. Για να ενημερώσει την μονάδα Issue το RS στο οποίο μπήκε η εντολή αλλά και για να την ενημερώσει ότι υπάρχει ελεύθερο RS χρησιμοποιείται η bus “Available”. Η σειρά προτεραιότητας αποθήκευσης μιας εντολής σε έναν RS είναι στατική (1>2>3), αντίθετα όμως η σειρά προτεραιότητας εκτέλεσης μιας εντολής ακολουθεί λογική round-robin.

Register File

| Σήμα | Πλάτος (bits) | Είδος | Περιγραφή |
|-----------|---------------|-------|---|
| ReadAddr1 | 5 | in | Address to be read |
| ReadAddr2 | 5 | in | Address to be read |
| CDBQ | 5 | in | CDB.Q Input |
| CDBV | 32 | in | CDB.V Input |
| Tag | 5 | in | Tag input |
| WrEn | 1 | in | Write enable to tag |
| AddrW | 5 | in | Address to Write the tag |
| Clk | 1 | in | Clock |
| Rst | 1 | in | Reset |
| DataOut1 | 32 | out | Data of Register with address ReadAddr1 |
| TagOut1 | 5 | out | Tag of Register with address ReadAddr1 |
| DataOut2 | 32 | out | Data of Register with address ReadAddr2 |
| TagOut2 | 5 | out | Tag of Register with address ReadAddr2 |

Τεκμηρίωση Λειτουργικότητας

Η Register File αποτελείται από 32 registers των 32 bit για τα data και παράλληλα άλλους 32 για τα tags. Διαθέτει δύο πόρτες ανάγνωσης από όπου λαμβάνονται και τα δεδομένα αλλά και το tag των δεδομένων που ζητούνται από τις δοσμένες διευθύνσεις. Η Register File γράφει τα tags μέσω της μονάδας Issue και παράλληλα αποθηκεύει τα δεδομένα CDB.V όταν βρεθεί κάποιο tag μέσα της ίσο με την είσοδο CDB.Q.

Common Data Bus

| Σήμα | Πλάτος (bits) | Είδος | Περιγραφή |
|-------------------|---------------|-------|--|
| QArithmetic | in | 5 | Q of Arithmetic unit |
| VArithmetic | in | 32 | V of Arithmetic unit |
| QLogical | in | 5 | Q of Logical unit |
| VLogical | in | 32 | V of Logical unit |
| QBuffer | in | 5 | Q of Buffer |
| VBuffer | in | 32 | V of Buffer |
| ArithmeticRequest | in | 1 | Request from Arithmetic unit |
| LogicalRequest | in | 1 | Request from Logical unit |
| BufferRequest | in | 1 | Request from Buffer |
| Clk | in | 1 | Clock |
| Rst | in | 1 | Reset |
| Qout | out | 5 | CDB.Q |
| Vout | out | 32 | CDB.V |
| GrantArithmetic | out | 1 | Grant to Arithmetic unit to write to CDB |
| GrantLogical | out | 1 | Grant to Logical unit to write to CDB |
| GrantBuffer | out | 1 | Grant to Buffer unit to write to CDB |

Τεκμηρίωση Λειτουργικότητας

Το CDB δέχεται τα ArithmeticRequest, LogicalRequest, BufferRequest και αποφασίζει ποιός θα γράψει στις εισόδους του σύμφωνα με την λογική round robin στέλνοντας στο αντίστοιχο module GrantArithmetic, GrantLogical ή GrantBuffer.

Issue

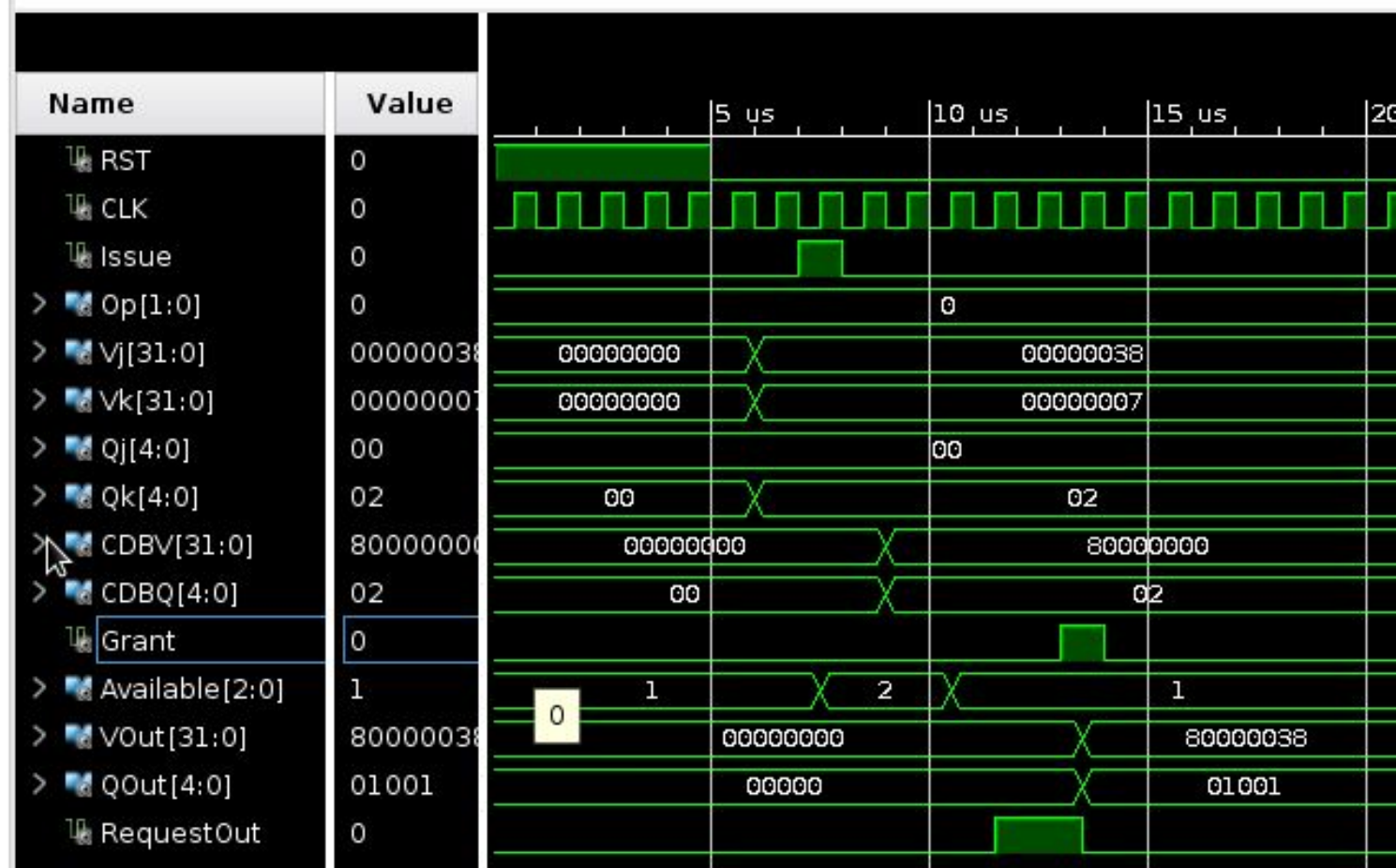
| Σήμα | Είδος | Πλάτος (bits) | Περιγραφή |
|---------------------|-------|---------------|---------------------------------|
| IssueIn | in | 1 | There are instructions from IF |
| FUType | in | 2 | Type of instruction |
| Fop | in | 2 | Operation to execute |
| Ri | in | 5 | Destination Register |
| Rj | in | 5 | Source Register 1 |
| Rk | in | 5 | Source Register 2 |
| RFReadAddr1 | out | 5 | Register to be read 1 |
| RFReadAddr2 | out | 5 | Register to be read 2 |
| RFTag | out | 5 | Tag to write at RFAddrW |
| RFAddrW | out | 5 | Address to write tag |
| RFWrEn | out | 1 | Register File Write Enable |
| Accepted | out | 1 | Instruction was Accepted |
| OpOut | out | 2 | Operation to FU |
| ArithmeticAvailable | in | 3 | Code of available Arithmetic RS |
| ArithmeticIssue | out | 1 | Write Enable of Arithmetic RS |
| LogicalAvailable | in | 3 | Code of available Logical RS |
| LogicalIssue | out | 1 | Write Enable of Logical RS |
| Clk | in | 1 | Clock |
| Rst | in | 1 | Reset |

Τεκμηρίωση Λειτουργικότητας

Το issue unit αποτελεί τον έλεγχο της έκδοσης εντολών προς τα RS και επιλέγει ποιο RS θα το χρησιμοποιήσει και πότε. Επιπλέον, γράφει στην Register File το tag του RS που χρησιμοποιείται ως έξοδος δεδομένων στον καταχωρητή προορισμού.

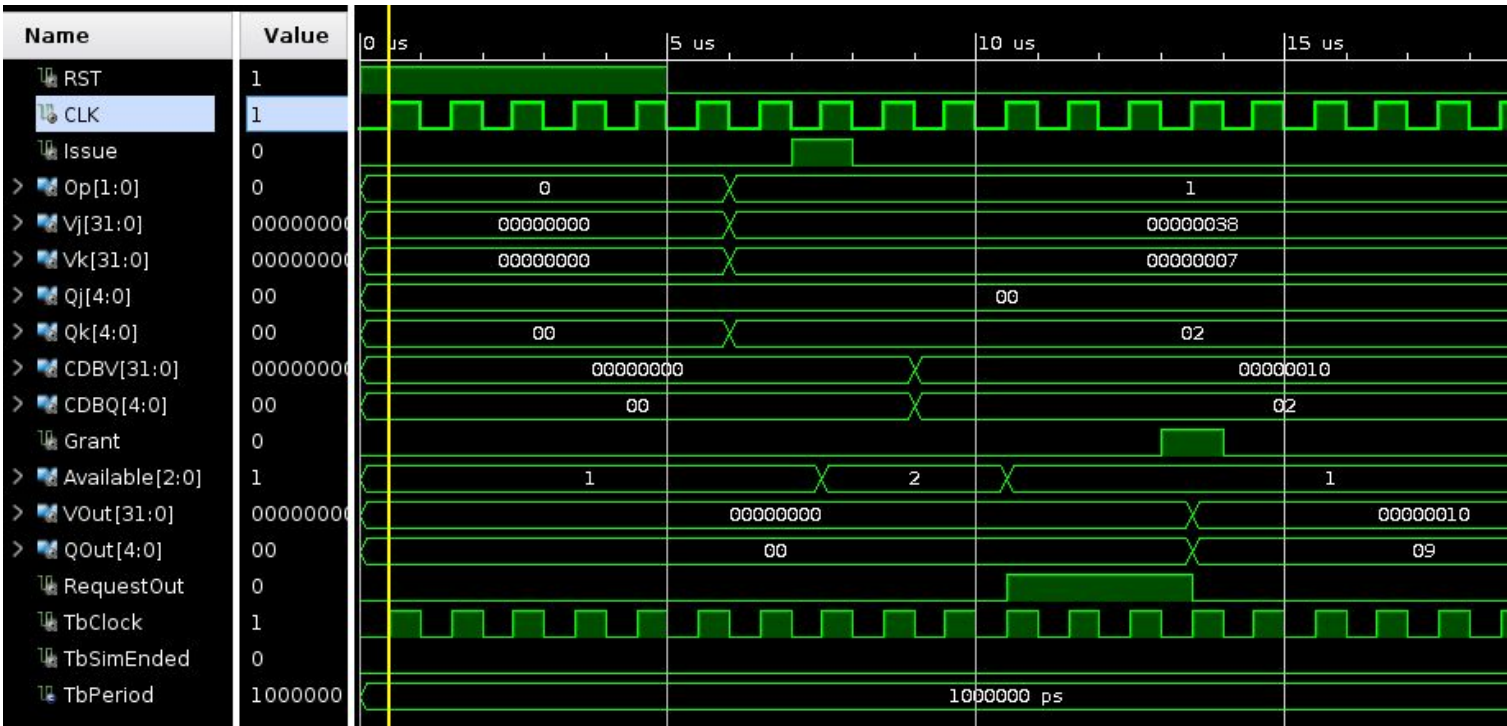
WaveForms:

Complete Arithmetic Unit



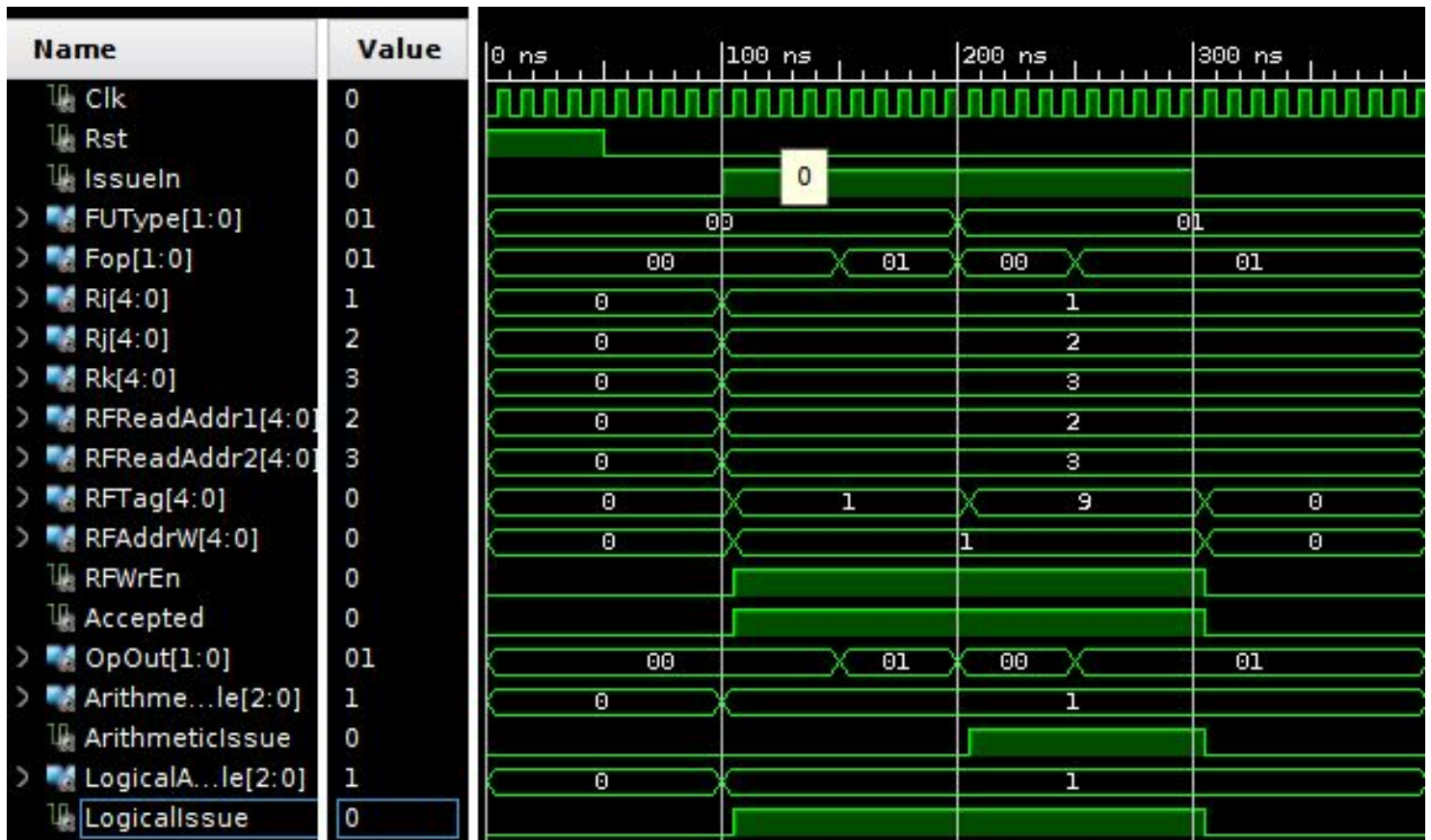
Παραπάνω βλέπουμε την κυματομορφή μιας ολοκληρωμένης Arithmetic Unit. Μπορούμε να παρατηρήσουμε ότι ενώ το σήμα Issue έρχεται αρκετά νωρίς, το σήμα request από την Arithmetic Unit αργεί να εμφανιστεί, αυτό συμβαίνει επειδή το Reservation Station περιμένει ένα δεδομένο από την CDB το οποίο έρχεται και δύο κύκλους πριν εμφανιστεί το σήμα request. Συγκεκριμένα 2 κύκλους, αφού η Arithmetic Functional Unit χρειάζεται 3 κύκλους για να ολοκληρώσει την εντολή, και ζητάει ακρόαση από το CDB έναν κύκλο πριν την ολοκλήρωσή της. Ακόμη παρατηρούμε το σήμα Available το οποίο και ενημερώνει την μονάδα Issue με το RS το οποίο επρόκειτο να λάβει την εντολή (σε περίπτωση που δεν υπάρχει ελεύθερη RS το σήμα Available είναι “000”). Τέλος μπορούμε να παρατηρήσουμε την έξοδο των δεδομένων μαζί με το tag του reserevation station που χρησιμοποιήθηκε να εμφανίζονται αμέσως μετά την λήψη του σήματος Grant από το CDB.

Complete Logical Unit



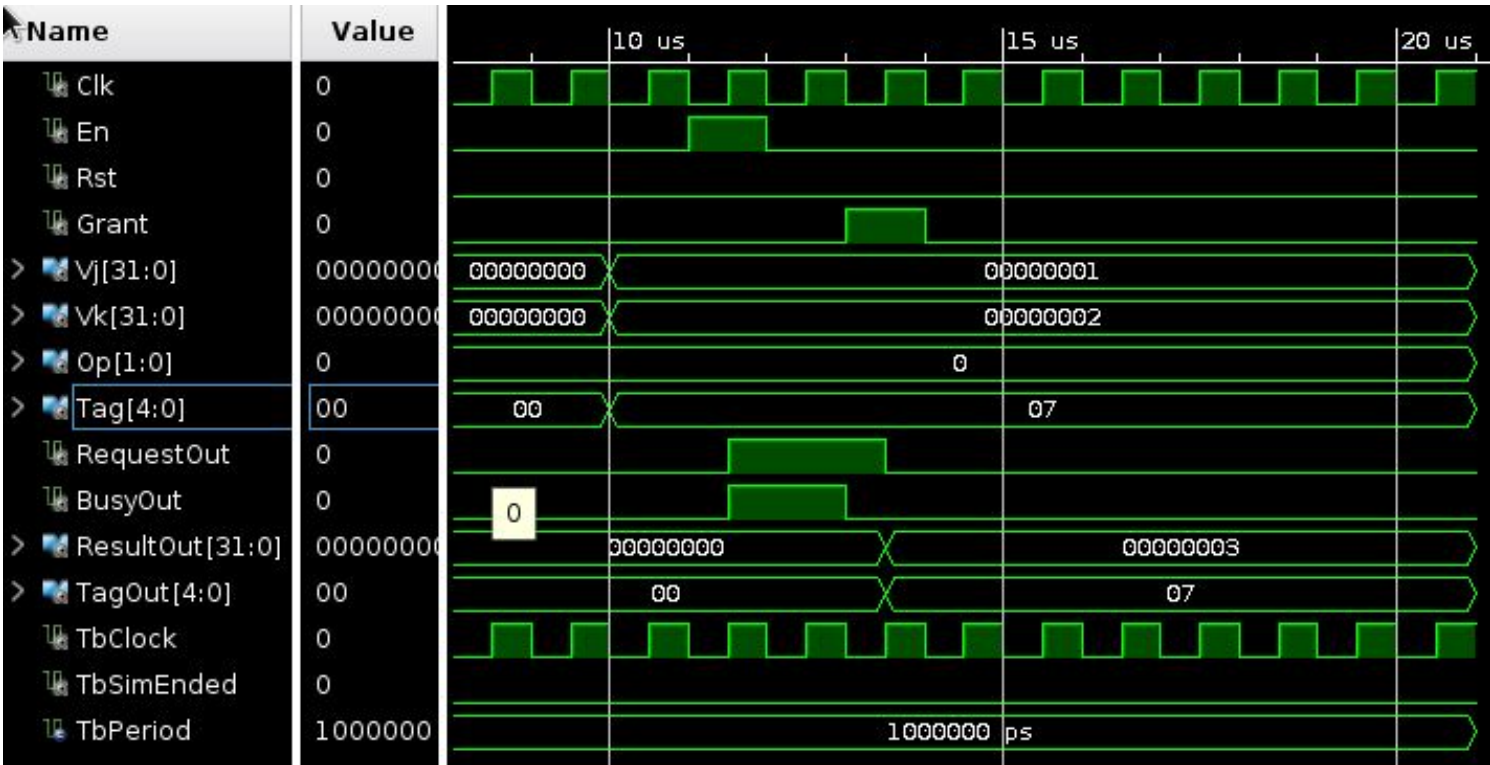
Παραπάνω βλέπουμε την κυματομορφή μιας ολοκληρωμένης Logical Unit. Η λειτουργία της είναι παρόμοια με την λειτουργία μιας ολοκληρωμένης Arithmetic Unit με μόνες διαφορές την πράξη που γίνεται, όπως και τον χρόνο να εμφανιστεί το σήμα request, αφού οι λογικές πράξεις παίρνουν 2 κύκλους ρολογιού για να ολοκληρωθούν.

Issue



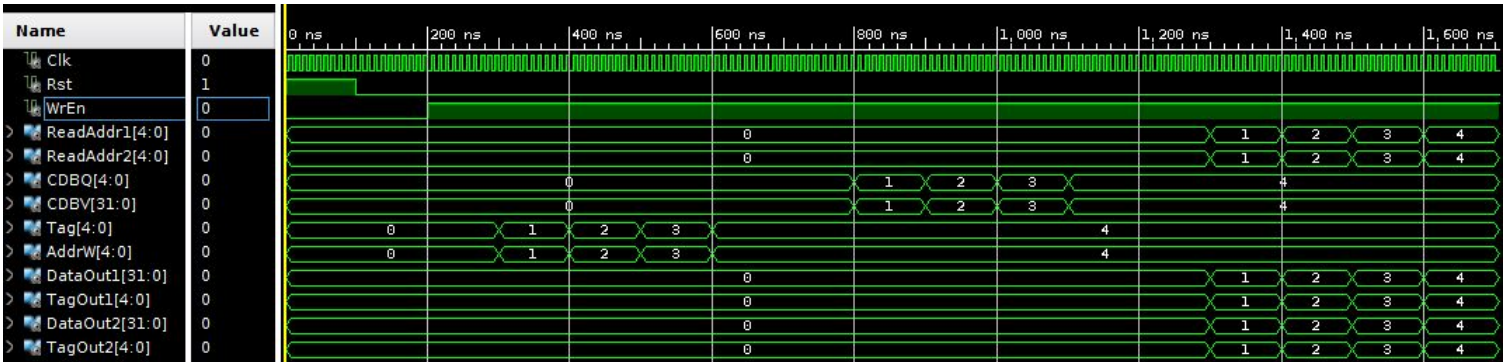
Παραπάνω διαφαίνονται οι κυματομορφές εισόδων εξόδων μίας πλήρους Issue μονάδας. Πιο συγκεκριμένα, μπορούμε να παρατηρήσουμε ότι η μονάδα μεταξύ των 100ns και 300ns δέχεται σήμα IssueIn και έτσι ενεργοποιείται η λειτουργία εγγραφής της εντολής στο κατάλληλο RS και παράλληλα η εγγραφή του tag του επιλεγμένου RS στον καταχωριτή προορισμού της Register File. Μεταξύ των 100-150 ns εκδίδεται η εντολή or R1, R2, R3, όπου σωστά επιλέγεται το Logical RS με tag 1 ($00001_2 = (1)_{10}$) το οποίο και γράφεται στο tag του R1 στην RF. Ομοίως συμβαίνει και στην αμέσως επόμενη εντολή and R1, R2, R3. Αμέσως μετά, εκδίδεται η εντολή add R1, R2, R3 όπου σωστά επιλέγεται το Arithmetic RS με tag 1 όπως διαφαίνεται στο σήμα RFTag ($01001_2 = (9)_{10}$) το οποίο και γράφεται στο tag του R1 στην RF. Ομοίως συμβαίνει και στην αμέσως επόμενη εντολή sub R1, R2, R3. Μετά τα 300ns το σήμα IssueIn απενεργοποιείται και αντίστοιχα το RFWrEn απενεργοποιείται.

Functional Unit



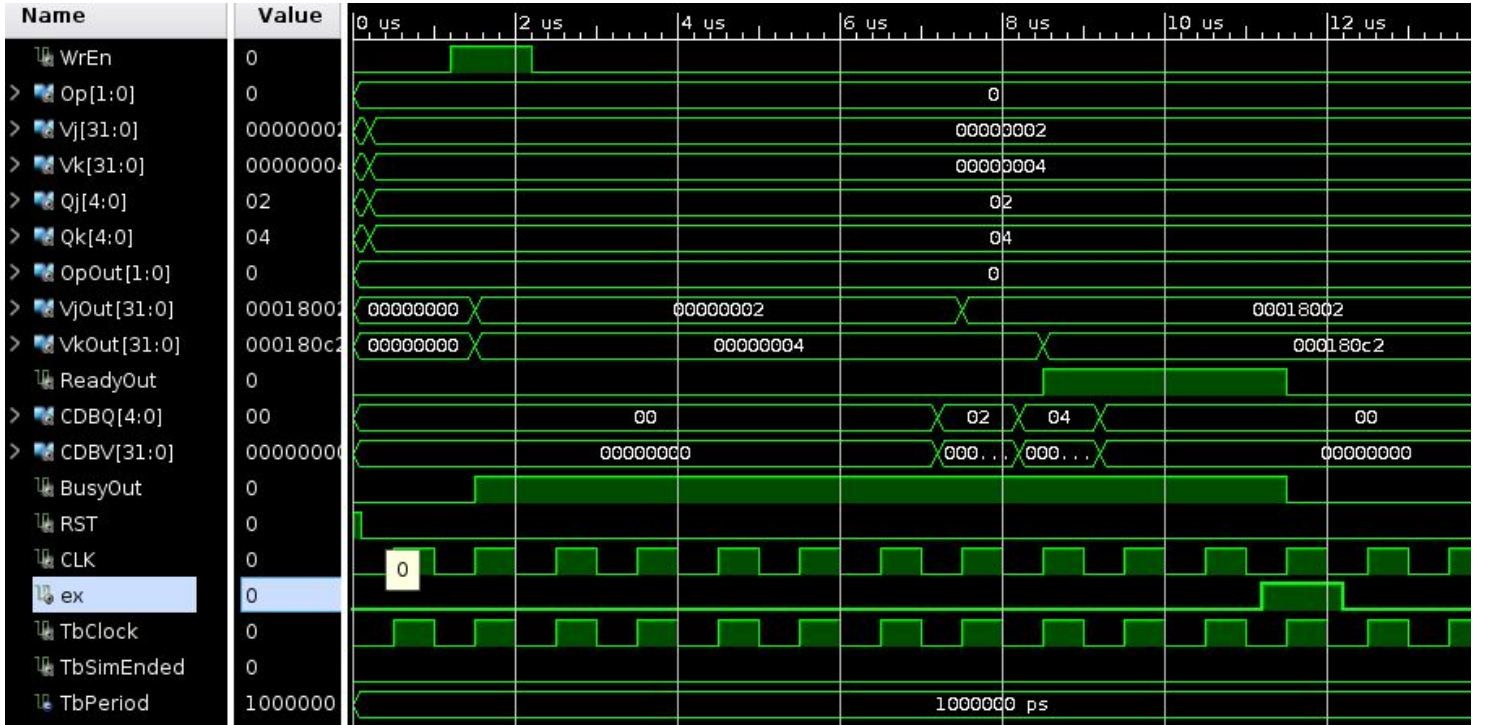
Παραπάνω βλέπουμε το simulation ενός logic functional unit. Το logic functional unit αποτελείται από μία μονάδα πράξεων και δύο pipeline registers. Αυτό συμβαίνει για την προσομοίωση ενός pipelined συστήματος 2 κύκλων. Η μονάδα χρησιμοποιεί ένα σήμα En για να δεχτεί καινούργιες εντολές. Μπορούμε να δούμε στην κυματομορφή ότι το σήμα request εμφανίζεται μόλις η μονάδα λάβει μια καινούρια εντολή, αφού το σήμα αυτό πρέπει εμφανίζεται ένα κύκλο πριν την έκδοση της εντολής. Ακόμη βλέπουμε ότι αφού η μονάδα λάβει το σήμα Grant, τα δεδομένα της πράξης εξάγονται στον επόμενο κύκλο. Με παρόμοιο τρόπο δουλεύει και το Arithmetic Functional Unit το οποίο όμως χρειάζεται 3 κύκλους για την ολοκλήρωση μιας εντολής, άρα και 2 κύκλους για την εμφάνιση του σήματος request.

Register File



Παραπάνω βλέπουμε την προσομοίωση της Register File όπου μπορούμε να παρατηρήσουμε ότι από τα 200ns και μετά γράφονται τα tags των R1, R2, R3 και R4 των οποίων οι τιμές των data αντικαθιστώνται από το CDB μετά τα 800 ns. Ακόμα μπορούμε να παρατηρήσουμε ότι οι τιμές που γράφτηκαν προηγουμένως διαβάζονται με επιτυχία μετά τα 1300ns.

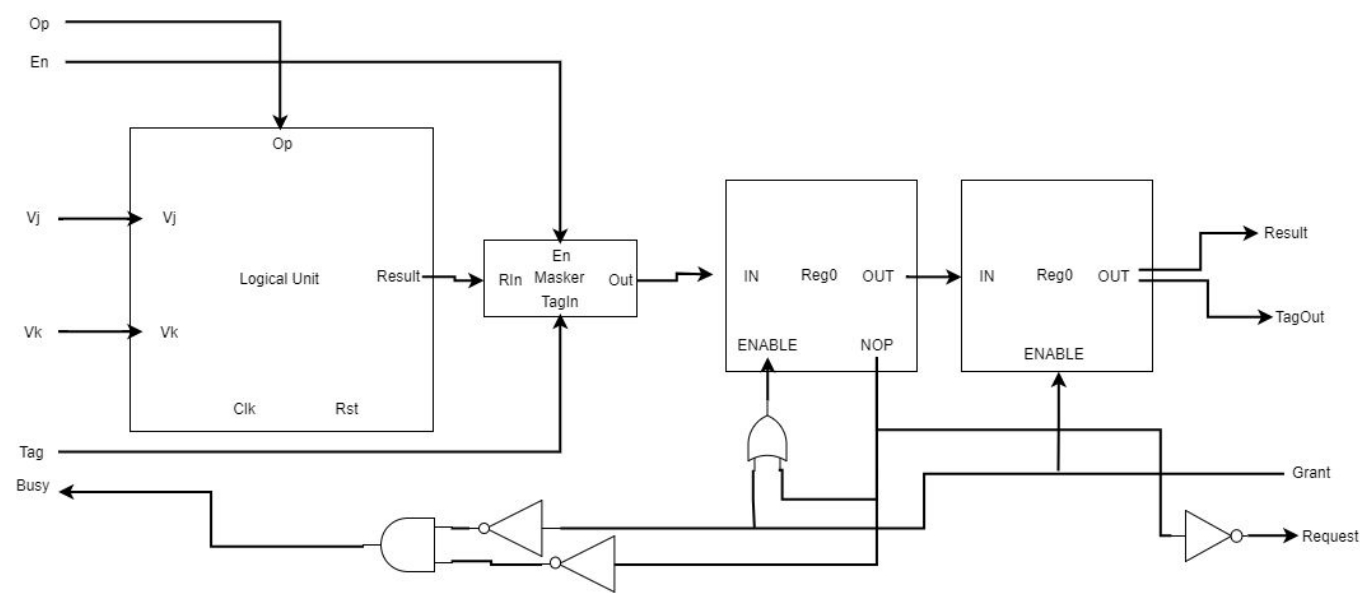
Reservation Station



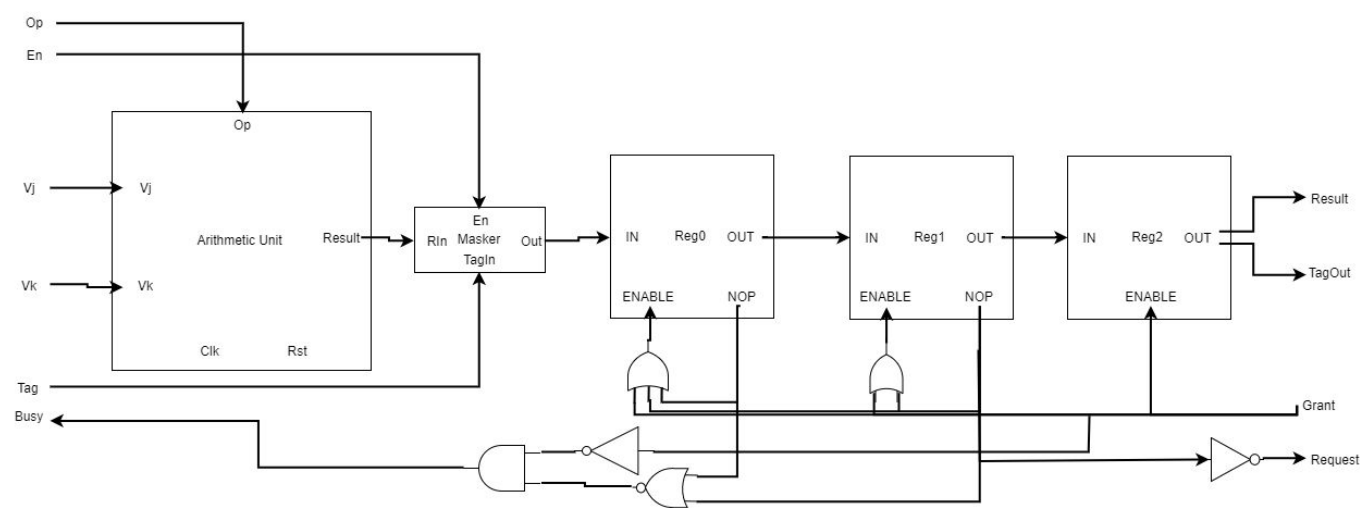
Στην κυματομορφή προσομοίωσης ενός reservation station υπάρχουν 3 σημεία ενδιαφέροντος. Πρώτο σημείο είναι η εμφάνιση του σήματος WrEn στο οποίο και καταγράφεται μια εντολή στο RS, μαζί με την καταγραφή της εντολής, το RS ορίζεται και ως Busy από το σήμα “BusyOut”. Βέβαια επειδή η εντολή περιμένει δεδομένα από το CDB το σήμα ReadyOut παραμένει ‘0’. Αυτό μας φέρνει στο δεύτερο σημείο ενδιαφέροντος όπου και είναι το χρονικό σημείο όπου το RS δέχεται δεδομένα από την CDB, μπορούμε να δούμε ότι μόνο τότε το σήμα ReadyOut γίνεται ‘1’. Τελευταίο στην λίστα μας είναι το σήμα BusyOut, το οποίο και μηδενίζεται μόλις έρθει το σήμα Ex (Executed), το οποίο και σηματοδοτεί την απελευθέρωση αυτού του Reservation Station.

Διαγράμματα:

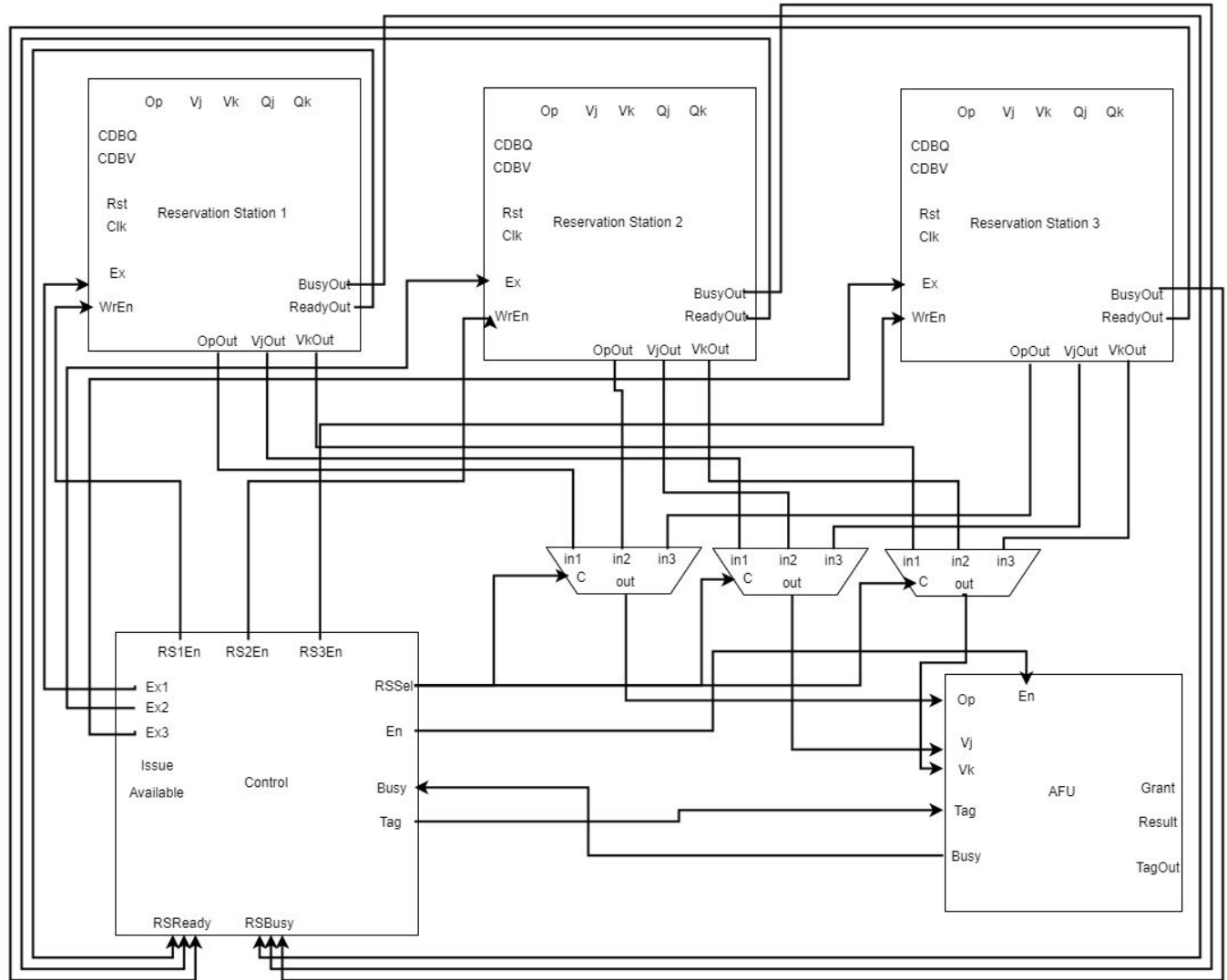
Logical Functional Unit



Arithmetic Functional Unit

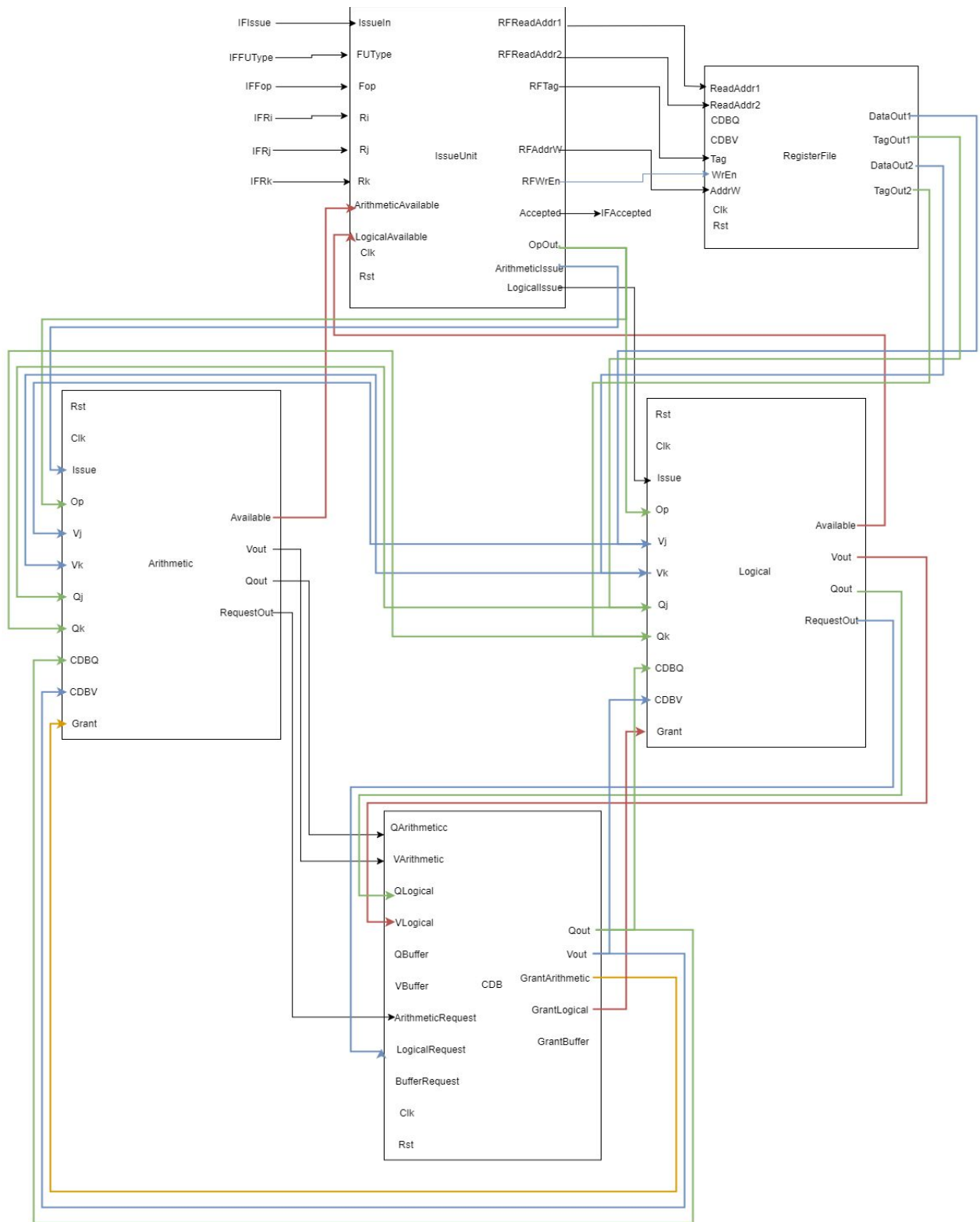


Arithmetic/Logical Functional Units with Reservation Station and Control



Το παραπάνω ενδεικτικό διάγραμμα αντιπροσωπεύει μια μονάδα αριθμητικών πράξεων. Παρόμοια υλοποιείται και μια μονάδα λογικών πράξεων με μόνες διαφορές την χρήση μιας LFU στην θέση της AFU και την χρήση δύο αντί τριών Reservation Station.

Complete System (Abstract)



Register File

