The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No 11, Nov 2025

Design and Implementation of a Buffer-Based High-Speed Data Transfer Router Using HDL

Nenavath Tharun, Dr. Bommidi Sridhar Banothu Bapuji ',
Salivoju Laharika
''UG Student, Department ofECE, Scient Institute of technology, Ibrahimpatnam, Telnngana, India, 'Department ofECE, Jawaharlal Nehru Technological University Hyderabad, Telaligana, India.
[3] *M. Tech, Dept ofECE, CVR college of Engineering, Ibrahimpatnam, Telangana, India*

రూపకల్పన మరియు అమలు యొక్క ఒక బఫర్ ఆధారితం హై-స్పీడ్ డేటా బదిలీ రూటర్ ఉపయోగించి హెచ్‌డిఎల్

నేనావత్ తరున్ , డాక్టర్ బొమ్మిడి శ్రీధర్, బానోతు బాపూజీ ,
సాలివోజు లహరిక
''యూజీ విద్యార్థి, ఇసిఇ విభాగం , సైయంట్ ఇన్స్టి ట్యూట్ ఆఫ్ టెక్నాలజీ, ఇబ్రహీంపట్నం , తెలంగాణ , '' భారతదేశం, 'ECE విభాగం , జవహర్‌లాల్ నెహ్రూ టెక్నలాజికల్ విశ్వవిద్యాలయం హైదరాబాద్, తెలగణ , భారతదేశం.
[3] M. Tech, Dept of ECE, CVR college of Engineering, Ibrahimpatnam, Telangana, India

Corresponding Author *: nenavaththarunnayak996J,qmai1.com

Abstract:

ఇది ప్రాజెక్ట్ ప్రగతిస్తుంది డిజైన్ మరియు అమలు యొక్క ఒక హార్డ్‌వేర్ డిస్క్రిప్షన్ లాంగ్వేజ్ (HDL) ఉపయోగించి బఫర్-ఆధారిత హై-స్పీడ్ డేటా ట్రాన్స్‌ఫర్ రౌటర్ , నెట్‌వర్క్-ఆన్-చిప్ (NoC) సిస్టమ్‌లలో త్రూపుట్‌ను మెరుగుపరచడం మరియు జాప్యాన్ని తగ్గించడం లక్ష్యంగా పెట్టుకుంది. ప్రతిపాదిత రౌటర్ ఆర్కిటెక్చర్ ప్రతి పోర్ట్ వద్ద FIFO బఫర్‌లను కలిగి ఉంటుంది, ఇది రౌండ్-రాబిన్ ఆర్బిట్రేషన్ ఫెయిర్ యాక్సెస్ కంట్రోల్ కోసం మెకానిజం, మరియు ఏకకాలంలో ఓవర్‌లోడ్ చేయబడిన కోడ్ డివిజన్ మల్టిపుల్ యాక్సెస్ (CDMA) క్రాస్‌బార్ స్విచ్ బహుళ-ఛానల్ కమ్యూనికేషన్. బఫరింగ్ యంత్రాంగం సమర్థవంతంగా బర్స్టీ ట్రాఫిక్‌ను నిర్వహిస్తుంది మరియు డేటా నష్టాన్ని నివారిస్తుంది, అయితే CDMA-ఆధారిత స్విచింగ్ గణనీయమైన ప్రాంతం లేదా పవర్ ఓవర్‌హెడ్ లేకుండా బ్యాండ్‌విడ్త్ వినియోగాన్ని పెంచుతుంది. ఈ డిజైన్ వెరిలాగ్ HDL లో వివరించబడింది , ఫంక్షనల్ వెరిఫికేషన్ కోసం Xilinx Vivado ని ఉపయోగించి అనుకరించబడింది మరియు FPGA అమలు కోసం సంశ్లేషణ చేయబడింది. ఆన్ ఒక ఆర్టిక్స్-7 పరికరం. పనితీరు విశ్లేషణ ప్రదర్శించారు తక్కువ జాప్యం, మెరుగైన నిర్గమాంశ, మరియు సమర్థవంతమైన వనరుల వినియోగం పోలిస్తే సాంప్రదాయ అన్‌బఫర్డ్ డిజైన్లు. ఈ పని అధిక-పనితీరు

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No 11, Nov 2025

గల SoC లకు అనువైన స్కేలబుల్ మరియు శక్తి-సమర్థవంతమైన రౌటర్ పరిష్కారాన్ని అందిస్తుంది. మరియు అధునాతన NoC ఆర్కిటెక్చర్‌లు.
కీలకపదాలు: VLSI రూటర్, నెట్‌వర్క్-ఆన్-చిప్ (NoC), వెరిలాగ్ HDL, పవర్ ఆప్టిమైజేషన్, పనితీరు, ప్రాంత సామర్థ్యం, డేటా కమ్యూనికేషన్, Xilinx Vivado , Zynq ప్లాట్‌ఫారమ్, ప్యాకెట్ స్విచింగ్, SoC డిజైన్.

1. పరిచయం

ఆధునిక **VLSI** మరియు **సిస్టమ్-ఆన్-చిప్ (SoC)** ఆర్కిటెక్చర్లలో, బహుళ ప్రాసెసింగ్ ఎలిమెంట్స్ (PEs) మధ్య వేగవంతమైన మరియు సమర్థవంతమైన కమ్యూనికేషన్ మొత్తం వ్యవస్థ పనితీరుకు కీలకం. సంఖ్యగా యొక్క భాగాలు ఒక చిప్ కొనసాగుతుంది కు పెరుగు, సాంప్రదాయ బస్సు ఆధారిత ఇంటర్‌కనెక్ట్ పద్ధతులు ఉన్నాయి చూపబడింది స్పష్టమైన లోపాలు - అవి ఉన్నాయి కష్టం కు స్కేల్, ధారపోయు కు సృష్టించు పనితీరు అడ్డంకులు, మరియు తరచుగా అధిక పనితీరు లేదా శక్తి-సున్నితమైన డిజైన్లలో ఆమోదయోగ్యమైన దానికంటే ఎక్కువ శక్తిని వినియోగిస్తాయి .
కు చిరునామా ఇవి సవాళ్లు, ది **నెట్‌వర్క్-ఆన్-చిప్ (ఎన్.ఓ.సి)** ఉదాహరణ కలిగి ఉంది ఉద్భవించింది గా ఒక స్కేలబుల్ మరియు సమర్థవంతమైన ప్రత్యామ్నాయం. ఒక లో ఎన్ఓసి , **రౌటర్లు** మరియు **లింకులు** భర్తీ చేయు పాతది పంచుకున్నారు బస్సు, కనెక్ట్ చేస్తోంది

సిస్టమ్ భాగాలు. ఈ విధానం బహుళ డేటా బదిలీలను ఒకే సమయంలో జరగడానికి అనుమతిస్తుంది, గణనీయంగా త్రూపుట్‌ను మెరుగుపరుస్తుంది మరియు రద్దీని తగ్గిస్తుంది. పాతది డిజైన్లు.
వద్ద ది గుండె యొక్క ప్రతి NOC లో **రౌటర్,** ఇది దాదాపుగా ఇలాగే పనిచేస్తుంది ఒక ట్రాఫిక్ కంట్రోలర్ కోసం ఆన్-చిప్ డేటా. దీని ప్రాథమిక పని ఒక పోర్ట్ నుండి ప్యాకెట్లను స్వీకరించడం మరియు వాటిని సరైన గమ్యస్థానానికి ఫార్వార్డ్ చేయడం. పోర్ట్. ఇది ప్రక్రియ ఉంటుంది అనేక సమన్వయ దశలు:

- **ప్యాకెట్ రిసెప్షన్** — డేటా ప్యాకెట్లు చేరుకోవడం వద్ద ఒకటి యొక్క ది రౌటర్లు ఇన్పుట్ పోర్టులు.
- **బఫరింగ్** — అవసరమైన అవుట్‌పుట్ పోర్ట్ బిజీగా ఉంటే , ఇన్‌కమింగ్ ప్యాకెట్ తాత్కాలికంగా బఫర్‌లో నిల్వ చేయబడుతుంది.
- **రూటింగ్ గణన** — ది రౌటర్ నిర్ణయిస్తుంది ఎక్కడ కు పంపండి ది ప్యాకెట్ తరువాత, అల్గోరిథంలను ఉపయోగించడం అటువంటి గా **నిర్ణయాత్మక XY తెలుగు in లో రూటింగ్** లేదా అనుకూల రూటింగ్ పద్ధతులు.
- **ఆర్బిట్రేషన్** — ఉంటే మరిన్ని కంటే ఒకటి ప్యాకెట్ కోరుకుంటున్నారు కు ఉపయోగం ది అదే

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No 11, Nov 2025

అవుట్పుట్ పోర్ట్, ది మధ్యవర్తి నిర్ణయిస్తాడు ఏది ఒకటి వెళ్తుంది ముందుగా, తరచుగా ఉపయోగించి **రౌండ్-రాబిన్** లేదా ప్రాధాన్యత ఆధారిత పద్ధతులు.

- **మారుతోంది** — ది రౌటర్లు అంతర్గత క్రాస్ బార్ కలుపుతుంది ది ఎంపిక చేయబడింది ఇన్పుట్ కు ది అవుట్పుట్, ప్యాకెట్ ముందుకు కదలడానికి వీలు కల్పిస్తుంది.
- **ప్యాకెట్ ఒకరి నుండి ఒకరికి వ్యాధి ప్రబలడం** — ది ప్యాకెట్ ఉంది పంపబడింది కు ది తరువాత

రౌటర్ లేదా దాని చివరి గమ్యస్థానం. అయితే ఇది అనిపిస్తుంది నేరుగా, విషయాలు అవ్వండి మరిన్ని సంక్లిష్టమైన కింద భారీ **ట్రాఫిక్.** చాలా ఉన్నప్పుడు ప్యాకెట్లు ఉన్నాయి పోటీ పడుతోంది కోసం ది అదే వనరులు, జాప్యాలు మరియు ప్యాకెట్ చుక్కలు చెయ్యవచ్చు సంభవిస్తాయి. ఇది **బఫర్లు** ప్లే చేసేది ఇక్కడే ముఖ్యమైన పాత్ర.

బఫర్లు చర్య గా తాత్కాలికమైన నిల్వ ప్రాంతాలు కోసం ప్యాకెట్లు అది సాధ్యం కాదు ఉండండి ఫార్వార్డ్ చేయబడింది వెంటనే. వారి ప్రధాన విధులు:

- **నిర్వహణ రద్దీ** — స్టోర్ ప్యాకెట్లు ఎప్పుడు అవుట్పుట్ పోర్టులు ఉన్నాయి బిజీగా, నిరోధించడం డేటా నష్టం.
- **ఆలస్యం తగ్గించడం** — పైప్లైనింగ్ను ప్రారంభించడం ద్వారా డేటా సజావుగా కదులుతూ ఉండండి , ఇక్కడ బహుళ ప్యాకెట్లు ఉంటాయి ఉన్నాయి ప్రాసెస్ చేయబడింది వివిధ దశలు ఏకకాలంలో.
- **బూస్టింగ్ సామర్థ్యం** — అనుమతించు నిరంతర డేటా ప్రవాహం, సరి ఎప్పుడు ట్రాఫిక్ ఉంది పగిలిపోయేవి .
- **మద్దతు ఇవ్వడం ప్రవాహం నియంత్రణ** — పని ఏదైనా క్రెడిట్ ఆధారిత లేదా కరచాలనం ప్రోటోకాల్లు కు బఫర్ ఓవర్ఫ్లోలు మరియు అండర్ఫ్లోలను నివారించండి .
- **బ్యాలెన్సింగ్ పనిభారాలు** — గ్రహించు ఆకస్మిక ట్రాఫిక్ వచ్చే చిక్కులు కాబట్టి అది ది రౌటర్ ఉంది సమయం వాటిని సమర్థవంతంగా ప్రాసెస్ చేయడానికి.

సారాంశంలో, రౌటర్లు డేటా ఎక్కడికి వెళుతుందో నిర్ధారిస్తాయి, బఫర్లు అక్కడికి **ఎంత సజావుగా** చేరుతుందో నిర్ధారిస్తాయి. కలిసి, అవి హై-స్పీడ్, విశ్వసనీయమైన ఆన్-చిప్ కమ్యూనికేషన్కు వెన్నెముకగా నిలుస్తాయి. బఫర్లు లేకుండా, ది అత్యంత వేగవంతమైన రౌటర్ గరిష్ట లోడ్ల కింద ఇబ్బంది పడుతుంది, చాలా లేకుండా బిజీగా ఉండే కూడలిలా వేచి ఉన్న కార్ల కోసం హోల్డ్ లేన్.

తెలివైన రూటింగ్, న్యాయమైన ఆర్బిట్రేషన్ మరియు బాగా నిర్వహించబడే బఫరింగ్లను కలపడం ద్వారా, ఆధునిక NoC రౌటర్లు నేటి సంక్లిష్టమైన VLSI వ్యవస్థలు కోరుకునే తక్కువ జాప్యం, అధిక **నిర్గమాంశ మరియు విశ్వసనీయతను సాధిస్తాయి.**

2. సాహిత్యం సర్వే

కుమార్ మరియు ఇతరులు (2002) స్కేలబుల్ను పరిచయం చేస్తూ స్ట్రక్చర్డ్ **నెట్వర్క్-ఆన్-చిప్ (NoC) ఆర్కిటెక్చర్ను ప్రతిపాదించిన మొదటి వారిలో ఉన్నారు.** మెష్ ఆధారిత టోపోలాజీలు మరియు

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No 11, Nov 2025

ప్యాకెట్-స్విచ్డ్ సాంప్రదాయ బస్సు-ఆధారిత వ్యవస్థల బ్యాండ్‌విడ్త్ మరియు స్కేలబిలిటీ పరిమితులను అధిగమించడానికి కమ్యూనికేషన్ . వారి పద్దతి సమితి ఒక బెంచ్‌మార్క్ కోసం NoC ని మూల్యాంకనం చేయడం పనితీరు నిబంధనలు యొక్క జాప్యం, నిర్గమాంశ మరియు విద్యుత్ సామర్థ్యం. ఈ పునాదులపై ఆధారపడి, జెర్రెగార్డ్ మరియు మహాదేవన్ (2006) సమర్పించారు ఒక సమగ్ర సర్వే యొక్క ఎన్.ఓ.సి. డిజైన్ పరిగణనలు, టోపోలాజీతో సహా ఎంపిక, ప్రవాహ నియంత్రణ మరియు రూటింగ్ వ్యూహాలు, పనితీరు మరియు మధ్య సమతుల్య ట్రేడ్-ఆఫ్‌ల అవసరాన్ని నొక్కి చెబుతాయి హార్డ్‌వేర్ ఖర్చు. ఈ ప్రాంతంలో ఆన్-చిప్ స్విచింగ్, బెల్ మరియు ఇతరులు (2001) మల్టీప్రాసెసర్ ఇంటర్‌కనెక్ట్‌ల కోసం **కోడ్-డివిజన్ మల్టిపుల్** యాక్సెస్ **(CDMA) ను ప్రవేశపెట్టారు** , ఆర్తోగోనల్‌ను ఉపయోగించుకున్నారు వ్యాప్తి చెందడం కోడ్‌లు కు ప్రారంభించు ఏకకాలిక సమాచార మార్పిడి పైగా ఒక పంచుకున్నారు మధ్యస్థం, తద్వారా మధ్యవర్తిత్వ సంక్లిష్టతను తగ్గిస్తుంది మరియు స్థిర జాప్యాన్ని హామీ ఇస్తుంది. నికోలిక్ మరియు ఇతరులు అల్. (2008) దీన్ని పొడిగించారు భావన ద్వారా రూపకల్పన స్కేలబుల్ సిడిఎంఎ పరిధీయ బస్సులు అది కనిష్టీకరించబడింది పిన్ లెక్కించబడుతుంది మరియు వైరింగ్ సంక్లిష్టత, మెరుగుదల మధ్య కమ్యూనికేషన్ ప్రాసెసింగ్ అంశాలు మరియు పరిధీయ పరికరాలు. లై మరియు అల్. (2004) CDMA ని టైమ్-డివిజన్ మల్టిపుల్ యాక్సెస్ తో కలిపి **CT-బస్ ఆర్కిటెక్చర్ ను** ప్రతిపాదించింది . (TDMA) నుండి పనితీరును మెరుగుపరచండి లో విజాతీయమైన ట్రాఫిక్ పరిసరాలు.

2. ప్రతిపాదించబడింది ఆర్కిటెక్చర్

ది ప్రతిపాదించబడింది విఎల్ఎస్ఐ రౌటర్ రెడీ ఉండండి రూపొందించబడిన తో ది అనుసరిస్తున్నారు

నిర్మాణ సంబంధమైన లక్షణాలు:

1.**విఎల్ఎస్ఐ రూటర్ తో బఫర్ కోసం వేగంగా డేటా బదిలీ:** ఇది బ్లాక్ చూపిస్తుంది ఒక ఉన్నత స్థాయి వీక్షణ యొక్క ఉత్తరం, దక్షిణం, తూర్పు మరియు పడమర అనే వివిధ దిశల మధ్య డేటా బదిలీని వేగవంతం చేయడంలో సహాయపడటానికి ఇంటిగ్రేటెడ్ బఫర్ ఉన్న VLSI రౌటర్. డిజిటల్ డేటా కోసం మీరు దీనిని నాలుగు-మార్గాల ట్రాఫిక్ ఖండనగా భావించవచ్చు. రౌటర్ మధ్యలో ఉంటుంది మరియు బఫర్ ఒక లాగా పనిచేస్తుంది. హోల్డింగ్ బే ఎక్కడ డేటా వేచి ఉంటుంది ముందు ఉండటం పంపబడింది బయటకు. ఇది బఫరింగ్ రద్దీని నివారిస్తుంది మరియు నిర్ధారిస్తుంది సాఫీగా సాగే ప్రవాహం, కేవలం ఇష్టం కార్లు క్యూలో నిల్చుని ఉండటం ఒక చిన్నది పార్కింగ్ ప్రాంతం ముందు ప్రవేశించడం ఒక బిజీగా త్రోవ.

2. **రూటర్ మరియు బఫర్ కనెక్షన్:** ఇక్కడ, రేఖాచిత్రం నేరుగా చూపించడం ద్వారా భావనను సులభతరం చేస్తుంది ఒక బఫర్ కనెక్ట్ చేయబడింది కు ఒక రౌటర్. ది బఫర్ ఉంది ముఖ్యంగా తాత్కాలికమైన నిల్వ కోసం ఇన్‌కమింగ్ లేదా అవుట్‌గోయింగ్ డేటా ప్యాకెట్లు, అయితే ది రౌటర్ ఎక్కడ

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No 11, Nov 2025

నిర్ణయించుకుంటుంది కు పంపండి వాటిని తరువాత. లో రోజువారీ నిబంధనల ప్రకారం, బఫర్‌ను ఒక చిన్న వెయిటింగ్ లాంజ్‌గా మరియు రౌటర్‌ను ప్రజలను సరైన కార్యాలయానికి నడిపించే రిసెప్షనిస్ట్‌గా ఊహించుకోండి.

3. **ఇన్‌పుట్-బఫర్ చేయబడింది రూటర్:** ఇది రేఖాచిత్రం ఉంది ఒక బిట్ మరిన్ని వివరణాత్మక. డేటా ప్రవేశిస్తుంది నుండి ది ఉత్తరం మరియు FIFO (ఫస్ట్-ఇన్-ఫస్ట్-అవుట్) క్యూలో నిల్వ చేయబడుతుంది — రాక క్రమంలో వ్యక్తులు వరుసలో ఉన్నట్లుగా. రూటింగ్ తర్కం నిర్ణయిస్తుంది ఏది అవుట్పుట్ పోర్ట్ ప్రతి ప్యాకెట్ తప్పక తీసుకోవడం. నుండి బహుళ ప్యాకెట్లు మేకావాలి కు వెళ్ళండి ది అదే మార్గం, మధ్యవర్తులు ఉన్నాయి ఉపయోగించబడింది కు నిర్ణయించు WHO పొందుతుంది కు వెళ్ళండి ముందుగా. ది మధ్యవర్తి పనిచేస్తుంది ఇష్టం ట్రాఫిక్ కాంతి, భరోసా ఇవ్వడం న్యాయము మరియు తప్పించుకోవడం ఘర్షణలు డేటా మార్గాలు.

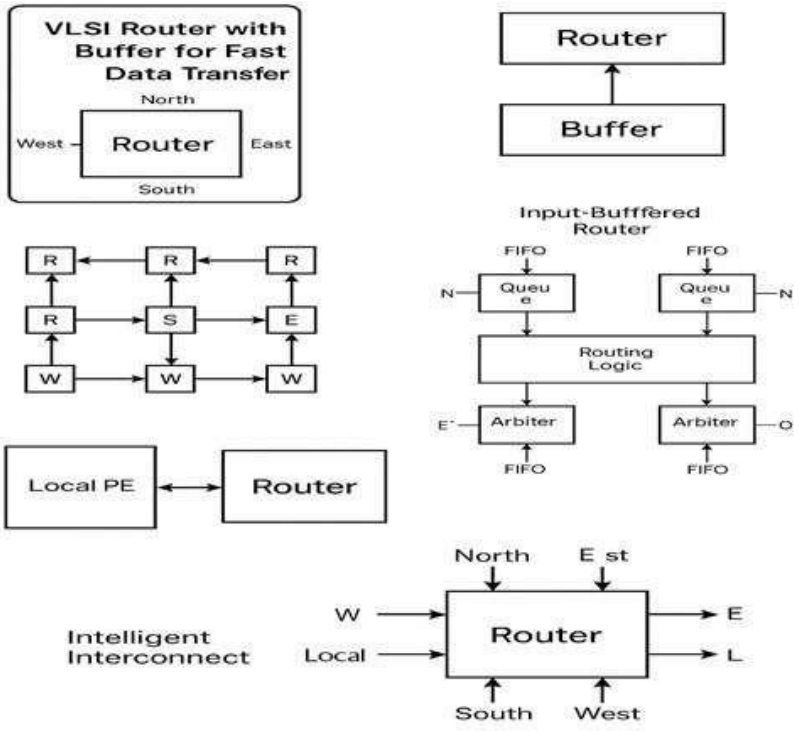
4. **ఇన్‌పుట్-బఫర్ చేయబడింది రూటర్:** ఇది రేఖాచిత్రం ఉంది ఒక బిట్ మరిన్ని వివరణాత్మక. డేటా ప్రవేశిస్తుంది నుండి ది ఉత్తరం మరియు FIFO (ఫస్ట్-ఇన్-ఫస్ట్-అవుట్) క్యూలో నిల్వ చేయబడుతుంది — రాక క్రమంలో వ్యక్తులు వరుసలో ఉన్నట్లుగా. రూటింగ్ తర్కం నిర్ణయిస్తుంది ఏది అవుట్పుట్ పోర్ట్ ప్రతి ప్యాకెట్ తప్పక తీసుకోవడం. నుండి బహుళ ప్యాకెట్లు మేకావాలి కు వెళ్ళండి ది అదే మార్గం, మధ్యవర్తులు ఉన్నాయి ఉపయోగించబడింది కు నిర్ణయించు WHO పొందుతుంది కు వెళ్ళండి ముందుగా. ది మధ్యవర్తి పనిచేస్తుంది ఇష్టం ట్రాఫిక్ కాంతి, భరోసా ఇవ్వడం న్యాయము మరియు తప్పించుకోవడం ఘర్షణలు డేటా మార్గాలు.

5. **రూటర్ నెట్‌వర్క్ టోపోలాజీ:** ఈ బాక్సుల గ్రిడ్ ఒకదానికొకటి అనుసంధానించబడిన బహుళ రౌటర్‌లను సూచిస్తుంది, ఇది ఒక నెట్‌వర్క్. ప్రతి "R" ఉంది ఒక రౌటర్, అయితే కొన్ని పెట్టెలు "W" అని లేబుల్ చేయబడింది కోసం పశ్చిమం, "ఇ" కోసం తూర్పు, మరియు "ఎస్" కోసం దక్షిణం — సూచిస్తూ దిశాత్మక పదవులు లో ది నెట్‌వర్క్. ఇది ఇలాంటి కు ఒక నగరం మ్యాప్ ఎక్కడ కూడళ్లు (రౌటర్లు) కనెక్ట్ చేయండి రోడ్లు (కమ్యూనికేషన్ లింకులు) కు సహాయం ట్రాఫిక్ (డేటా) ప్రవాహం మొత్తం నగరం అంతటా (చిప్).

6. **స్థానికం పిఇ కు రూటర్ కనెక్షన్:** ఇది రేఖాచిత్రం చూపిస్తుంది ఎలా ఒక స్థానిక ప్రాసెసింగ్ మూలకం (పిఇ)
— ముఖ్యంగా "మెదడు" లేదా కంప్యూటింగ్ కోర్ — కలుపుతుంది కు ఒక రౌటర్. ది పిఇ ఉత్పత్తి చేస్తుంది డేటా దానికి ప్రయాణించాలి ఇతర భాగాలు చిప్, మరియు రౌటర్ అనేది ది ఆ డేటాను పంపడానికి మరియు స్వీకరించడానికి గేట్‌వే. PE ని ఇల్లుగా మరియు రౌటర్‌ను స్థానిక పోస్టాఫీసుగా భావించండి. అన్ని అవుట్‌గోయింగ్ మరియు ఇన్‌కమింగ్ లెటర్‌లు (డేటా) దాని గుండా వెళతాయి.

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No 11, Nov 2025

7. **తెలివైన ఇంటర్‌కనెక్ట్:** చివరగా, ది చివరిది రేఖాచిత్రం చూపిస్తుంది ఒక మరిన్ని సంక్లిష్టమైన రౌటర్ తో నుండి ఇన్‌పుట్‌లు బహుళ ఆదేశాలు — పశ్చిమం, తూర్పు, ఉత్తరం, దక్షిణ, మరియు స్థానిక. ఇది పిలిచారు "తెలివైన" ఎందుకంటే ఇది డేటా ప్రయాణించడానికి అత్యంత సమర్థవంతమైన మార్గాన్ని నిర్ణయించగలదు, డైనమిక్‌గా పనిచేసే స్మార్ట్ GPS వ్యవస్థ లాగా దారి మళ్లింపులు ట్రాఫిక్ ఆధారిత ఆన్ పరిస్థితులు. ఇది నిర్ధారిస్తుంది ది అతి చిన్నది ప్రయాణం సమయం మరియు చిప్ నెట్‌వర్క్ లోపల డేటా రద్దీని నివారిస్తుంది.



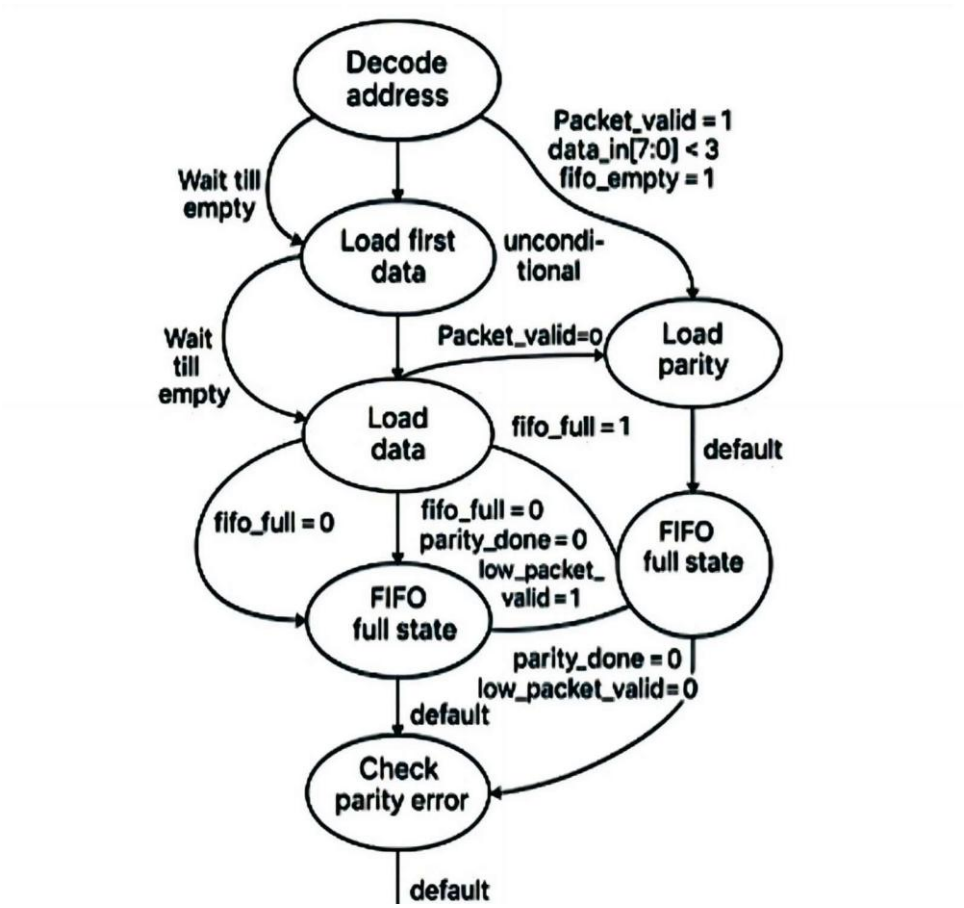
చిత్రం 1. ప్రతిపాదించబడింది ఆర్కిటెక్చర్

4. రూటర్ ఎఫ్ఎస్ఎం రాష్ట్రం రేఖాచిత్రం

రౌటర్ FSM అనేది స్మార్ట్ పోస్ట్ ఆఫీస్ లోపల ట్రాఫిక్ కంట్రోలర్ లాంటిది. ప్యాకేజీ (డేటా ప్యాకెట్) వస్తుంది, ఇది నియంత్రిక నిర్ణయిస్తుంది ఎక్కడ అది తప్పక వెళ్ళండి, ఎప్పుడు అది చెయ్యవచ్చు

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No 11, Nov 2025

ఉండండి నిల్వ చేయబడిన, మరియు ఎప్పుడు అది పంపవచ్చు — ఏ ప్యాకేజీ పోకుండా లేదా కలపబడకుండా చూసుకుంటూనే. సరైన డెలివరీ కౌంటర్ (అవుట్‌పుట్ పోర్ట్)ను గుర్తించడానికి ఇది ముందుగా లేబుల్ (డీకోడ్ చిరునామా)ను చదువుతుంది. నిల్వ బిన్ (FIFO)లో స్థలం ఉంటే, అది ప్యాకేజీని వరుసలో ఉంచుతుంది; బిన్ నిండి ఉంటే, స్థలం ఖాళీ అయ్యే వరకు అది వేచి ఉంటుంది. ఒకసారి ది ప్యాకేజీ ఉంది నిల్వ చేయబడిన, అది కూడా తనిఖీలు ది సీల్ (సమానత్వం తనిఖీ చేయండి) కు నిర్ధారించు ఏమీ లేదు ఉంది దెబ్బతిన్న రవాణా. కేవలం ఇష్టం ఒక మంచిది పోస్ట్ కార్యాలయం మేనేజర్ ఉంచుతుంది ది మెయిల్ ప్రవహించే సజావుగా, ది రూటర్ FSM ఉంచుతుంది డిజిటల్ డేటా కదులుతోంది క్రమబద్ధమైన, ఆన్ సమయం, మరియు లోపల దోష రహితం ఒక నెట్‌వర్క్.



The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No 11, Nov 2025

చిత్రం 2. రూటర్ ఎఫ్ఎస్ఎం రాష్ట్రం రేఖాచిత్రం

కార్యాచరణ:

ది కార్యాచరణ యొక్క ఇది ఎఫ్ఎస్ఎం కోసం FIFO-ఆధారితం ప్యాకెట్ నిర్వహణ చెయ్యవచ్చు ఉండండి వివరించబడింది దశలవారీగా గా
క్రిందివి :

1. చిరునామా డీకోడింగ్: ది ఎఫ్ఎస్ఎం ప్రారంభమవుతుంది లో ది డీకోడ్ చేయండి చిరునామా రాష్ట్రం, ఎక్కడ అది తనిఖీలు ప్యాకెట్ యొక్క గమ్యస్థాన చిరునామా (డేటా [7:0] 3 లో) మరియు ధృవీకరిస్తుంది అది ది ప్యాకెట్ ఉంది చెల్లుబాటు అయ్యే (ప్యాకెట్ చెల్లుబాటు అయ్యే = 1) మరియు అది ది సంబంధిత FIFO తెలుగు in లో ఉంది ఖాళీ. ఇది నిర్ధారిస్తుంది అది ఇన్‌కమింగ్ డేటా ఉంది దర్శకత్వం వహించిన కు సరైన FIFO అవుట్‌పుట్ ఛానల్.
2. మొదటి డేటా బైట్‌ను లోడ్ చేస్తోంది: మొదటి డేటాను లోడ్ చేయడానికి కదులుతుంది, ప్యాకెట్ యొక్క మొదటి బైట్‌ను నిల్వ చేస్తుంది ది FIFO. వేచి ఉంది కోసం FIFO తెలుగు in లో సంసిద్ధత (ఖాళీ రాష్ట్రం) ముందు అంగీకరించడం మరిన్ని డేటా.
3. మిగిలిన డేటాను లోడ్ చేస్తోంది: లో లోడ్ డేటా స్థితి, తదుపరి బైట్లు యొక్క ప్యాకెట్ అంటే లోడ్ చేయబడింది ది FIFO తెలుగు in లో నిరంతరం. ఉంటే ది FIFO తెలుగు in లో అవుతుంది పూర్తి (ఫైఫో పూర్తి = 1), ది ఎఫ్ఎస్ఎం తాత్కాలికంగా తరలిస్తుంది ది FIFO తెలుగు in లో పూర్తి రాష్ట్రం మరియు వేచి ఉంటుంది కోసం స్థలం కు అందుబాటులోకి వస్తాయి.
4. పూర్తి FIFOను నిర్వహించడం: FIFO పూర్తి స్థితిలో, fifo_full 0 అయ్యే వరకు సిస్టమ్ డేటా రైటింగ్‌ను పాజ్ చేస్తుంది. మళ్ళీ. ఒకసారి ఉంది స్థలం, అది రెజ్యూమ్‌లు లోడ్ అవుతోంది ది మిగిలిన డేటా.
5. పారిటీని లోడ్ చేస్తోంది బైట్: ఎప్పుడు అన్నీ ప్యాకెట్ డేటా ఉంది అందుకుంది (ప్యాకెట్ చెల్లుబాటు అయ్యే = 0), ది ఎఫ్ఎస్ఎం కదలికలు పారిటీ బైట్‌ను నిల్వ చేయడానికి పారిటీని లోడ్ చేయడానికి. ఈ పారిటీ బైట్ ప్రసార లోపాలను తనిఖీ చేయడానికి ఉపయోగించబడుతుంది .
6. పారిటీ చెకింగ్: లెక్కించిన పారిటీని పోల్చి, FSM చెక్ పారిటీ ఎర్రర్ స్థితిలోకి ప్రవేశిస్తుంది. నుండి ది అందుకుంది డేటా తో ది అందుకుంది సమానత్వం బైట్. ఉంటే వారు మ్యాచ్, లేదు లోపం ఉంది నివేదించబడింది; అవి భిన్నంగా ఉంటే, ఒక దోష సంకేతం పెంచారు.
7. తిరిగి వస్తున్నారు ఐడిల్: పారిటీ వెరిఫికేషన్ తర్వాత, FSM డీకోడ్ అడ్రస్‌కు తిరిగి రీసెట్ అవుతుంది తదుపరి ప్యాకేజీ కోసం వేచి ఉండండి.

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No 11, Nov 2025

5. ఫలితం

ఆర్‌టిఎల్ స్కీమాటిక్

ఇది డిజైన్ అంటే ఇష్టం ఒక చక్కగా నిర్వహించబడిన పోస్టల్ సార్టింగ్ సెంటర్, కానీ కోసం డిజిటల్ డేటా. ఇది నిర్మించబడింది నుండి ఆరు ప్రధాన "విభాగాలు": FSM రూటర్, రూటర్ రిజిస్టర్, సింక్ యూనిట్ మరియు మూడు FIFO బఫర్‌లు —ప్రతి అవుట్‌పుట్ పాత్‌కు ఒకటి.

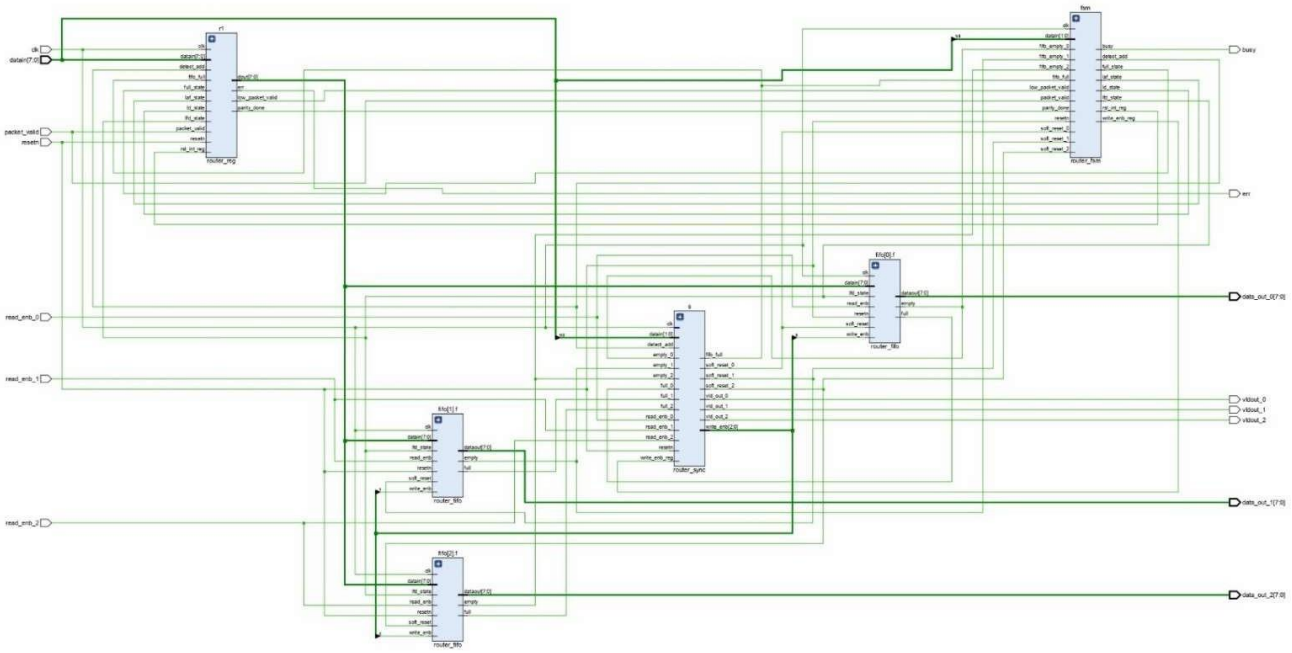
FSM రూటర్ (router_fsm) — దీన్ని ట్రాఫిక్ కంట్రోలర్‌గా భావించండి. ప్రతి ఇన్‌కమింగ్ డేటా ఎక్కడ ఉండాలో ఇది నిర్ణయిస్తుంది ప్యాకెట్ ఉండాలి వెళ్ళండి, ఉత్పత్తి చేస్తుంది ది నియంత్రణ సంకేతాలు, మరియు ఉంచుతుంది నడుస్తున్న ప్రతిదీ కుడి ఆర్డర్. ఇది చెబుతుంది ది రిజిస్టర్లు ఎప్పుడు కు సంగ్రహించు డేటా మరియు నిర్దేశిస్తుంది ప్రతి FIFO తెలుగు in లో ఎప్పుడు కు స్టోర్ లేదా విడుదల చేయండి.

రూటర్ రిజిస్టర్ (router_reg) — ఇది ది రిసెప్షన్ డెస్క్. ఎప్పుడైనా ప్యాకెట్ వస్తుంది, అది నిల్వ చేస్తుంది ఇన్‌కమింగ్ స్థితి, డేటా, మరియు సమానత్వం బిట్స్. కానీ అది మాత్రమే నవీకరణలు దాని రికార్డులు ఎప్పుడు ది ఎఫ్ఎస్ఎం ఇస్తుంది ఆకుపచ్చ తేలికైనది - నిర్ధారించుకోవడం ఏమీ లేదు పొందుతుంది ఓవర్‌రైట్ చేయబడింది ద్వారా ప్రమాదం.

మూడు FIFO తెలుగు in లో బఫర్‌లు (రౌటర్_ఫిఫో0, రౌటర్_ఫిఫో1, రౌటర్_ఫిఫో2) — ఇవి ఉన్నాయి ఇష్టం పట్టుకోవడం కోసం బేలు అవుట్‌గోయింగ్ మెయిల్. ప్రతి FIFO కి లింక్ చేయబడింది ఒక అవుట్‌పుట్ పోర్ట్ మరియు అది సిద్ధమయ్యే వరకు తాత్కాలికంగా డేటాను నిల్వ చేస్తుంది కు ఉండండి పంపబడింది బయటకు. ఏ FIFO పొందుతుంది ది డేటా ఆధారపడి ఉంటుంది ది FSMలు రూటింగ్ నిర్ణయం.

సమకాలీకరణ యూనిట్ (రూటర్_సింక్) — ఇది ఉంది ది దూత అది నిర్ధారిస్తుంది మృదువైన కమ్యూనికేషన్ మధ్య FSM మరియు FIFO లు. ఇది అన్నీ ఉంచుతుంది భాగాలు అలా సమకాలీకరించు ది సింగిల్ ఇన్‌పుట్ పోర్ట్ డబ్బా పంపండి డేటా శుభ్రంగా మరియు విశ్వసనీయంగా ఏదైనా యొక్క ది మూడు అవుట్‌పుట్ పోర్ట్‌లు ఎటువంటి గందరగోళం లేకుండా.

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No 11, Nov 2025

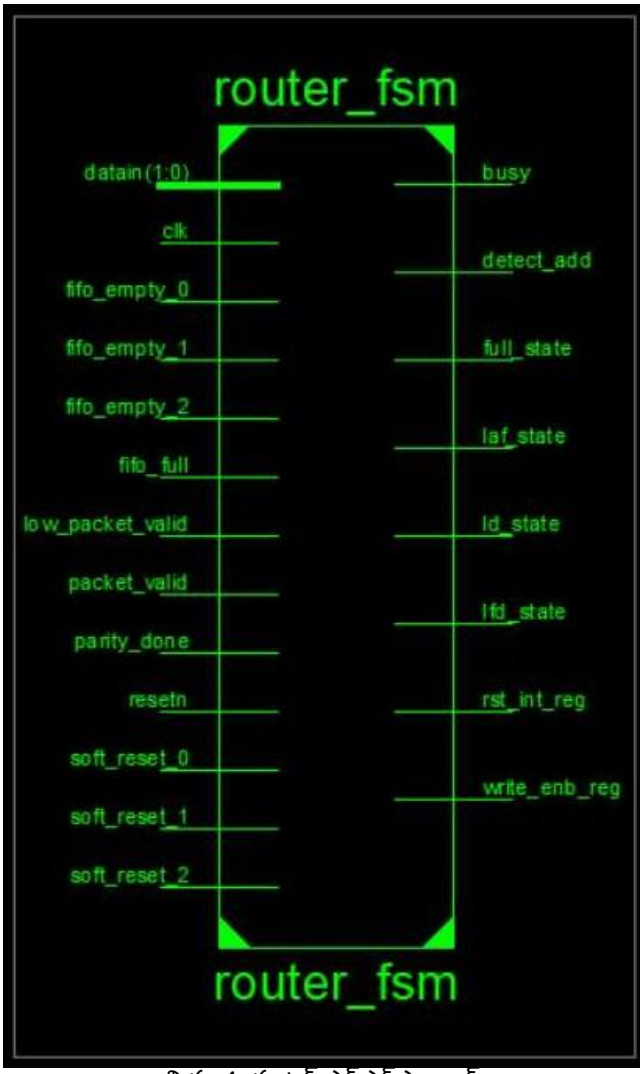


చిత్రం .3. ఆర్‌టిఎల్ స్కీమాటిక్ రేఖాచిత్రం యొక్క ఒక రౌటర్

రూటర్ ఎఫ్ఎస్ఎం బ్లాక్

router_fsm అనేది బిజీగా ఉండే డిజిటల్ కూడలిలో చీఫ్ ట్రాఫిక్ ఆఫీసర్ లాంటిది. ఇది వాస్తవానికి ది డేటా స్వయంగా — బదులుగా, 1టన్ గడియారాలు ప్రతిదీ, చేస్తుంది నిర్ణయాలు, మరియు సంకేతాలు ఇతర బ్లాక్స్ సరిగ్గా సరైన సమయంలో చర్య తీసుకోండి.

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No 11, Nov 2025



చిత్రం 4. రూటర్ ఎఫ్ఎస్ఎం బ్లాక్

కార్యాచరణ:

ఇన్‌పుట్‌లు (ఎడమ **వైపు)**

డేటాఇన్ (1:0) —› ఇవి ఇష్టం ది గమ్యస్థానం ట్యాగ్‌లు ఆన్ పొట్లాలు. ది ఎఫ్ఎస్ఎం వాటిని చదివి వినిపిస్తుంది బొమ్మ ఏ అవుట్‌పుట్ పోర్ట్‌ను బయటకు తీయాలి డేటా ఉండాలి కు పంపబడింది.

fifo_empty_0 / fifo_empty_1 / fifo_empty_2 ఇవి జెండాలు చెప్పు ది ఎఫ్ఎస్ఎం లేదో ప్రతి అవుట్‌పుట్ నిల్వ బిన్ (FIFO) ఖాళీగా ఉంది.

fifo_full ఇది ఉంది ఇష్టం ఒక "గిడ్డంగి అంటే పూర్తి" సంకేతం. ఉంటే ది FIFO తెలుగు in లో ఉంది

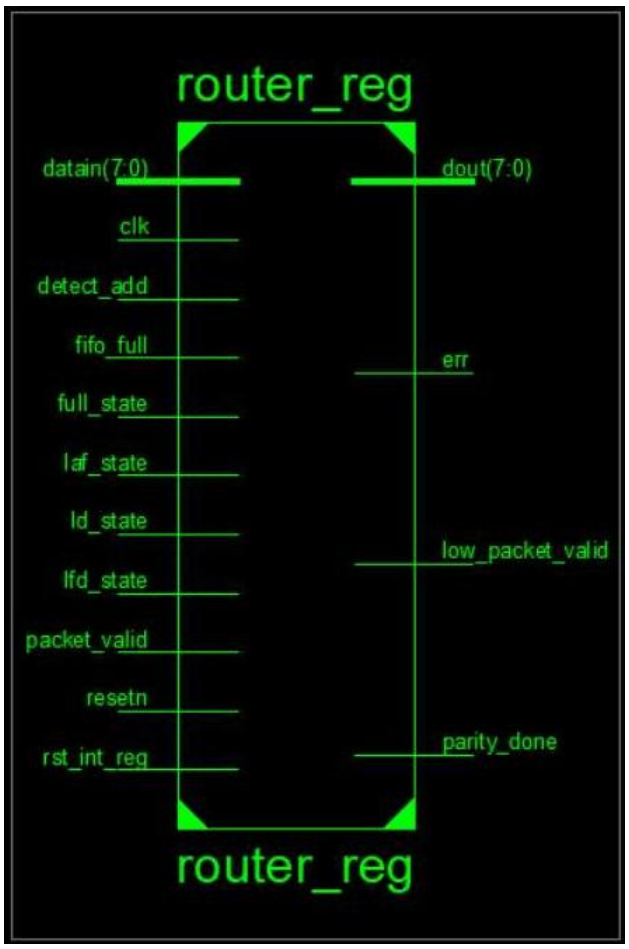
The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No 11, Nov 2025

పూర్తి, ది ఎఫ్ఎస్ఎం తెలుసు అది అవసరాలు అక్కడికి మరింత డేటాను పంపడాన్ని పాజ్ చేయడానికి. తక్కువ acket_valid / ప్యాకెట్_చెల్లుబాటు అయ్యేది —+ ఇవి ఇలా ఉంటాయి "కొత్త రవాణా ఉంది వచ్చింది" సంకేతాలు. వారు చెబుతారు FSM ఎప్పుడు ఇన్‌కమింగ్ డేటా ఉంది సిద్ధంగా ఉంది ప్రక్రియ.

పారిటీ_డన్ — + ఇది ఒక "నాణ్యత తనిఖీ ఆమోదించబడింది" ఫ్లాగ్. ఇది చెబుతుంది ది ఎఫ్ఎస్ఎం ది డేటా పారిటీ చెక్ పూర్తయింది.
resetn / soft_reset_0/1/2 ఇవి ఉన్నాయి ది అత్యవసర పరిస్థితి ఆపండి లేదా పునఃప్రారంభించు బటన్లు కోసం ది ఎఫ్ఎస్ఎం లేదా నిర్దిష్ట FIFO ల కోసం.
క్లిక్ చేయండి —+ ది హృదయ స్పందన అది ఉంచుతుంది ప్రతిదీ కదులుతోంది లో సమకాలీకరించండి.

అవుట్‌పుట్‌లు (కుడి వైపు)
బిజీగా —› అ గుర్తు కు ది బయట ప్రపంచం మాట్లాడుతూ, "పట్టుకోండి ఆన్, నేను సరిగ్గా పని చేస్తోంది ఇప్పుడు." **డిటెక్ట్_యాడ్** —+ ది క్షణం ది ఎఫ్ఎస్ఎం గుర్తిస్తుంది ది గమ్యస్థాన చిరునామా యొక్క ది ప్యాకెట్. **పూర్తి_స్థితి / లాఫ్_స్టేట్ —›** అంతర్గత రాష్ట్రాలు చెప్పడం ది వ్యవస్థ ఉంటే నిల్వ ఉంది పూర్తి లేదా దాదాపు నిండింది.
ld_స్టేట్ / lfd_స్టేట్ —› రాష్ట్రాలు కోసం ఎప్పుడు ది ఎఫ్ఎస్ఎం లోడ్ చేయాలి డేటా లేదా లోడ్ ది ముందుగా డేటా పదం FIFO లోకి .
మొదటి_ఇంట్_రెగ్ —+ ఆర్డర్లు కు రీసెట్ చేయండి ది అంతర్గత నమోదు చేయండి లో ది రౌటర్.
వ్రాయండి_enb_reg —+ రూటర్ రిజిస్టర్‌కి, "ముందుకు సాగండి, ఈ ఇన్‌కమింగ్ డేటాను క్యాప్చర్ చేయండి" అని చెప్పే ఆకుపచ్చ లైట్.
రూటర్ నమోదు చేయండి బ్లాక్
ది రౌటర్_రెగ్ ఉంది ఇష్టం ది రిసెప్షన్ డెస్క్ లో మా డిజిటల్ పోస్ట్ కార్యాలయం.
ఎప్పుడు ఒక కొత్త రవాణా డేటా వస్తుంది, ఇది ఉంది ది ముందుగా ఆపండి. ది రిసెప్షనిస్ట్ (రూటర్_రెగ్) జాగ్రత్తగా నమోదు చేయండి అన్నీ ది వివరాలు, ఉంచుతుంది వాటిని సురక్షితమైన, మరియు మాత్రమే పాస్‌లు వాటిని ముందుకు ఎప్పుడు ది మేనేజర్ (FSM) ఇదే సరైన సమయం అని చెబుతోంది.

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No 11, Nov 2025



చిత్రం 5. రూటర్ రెగ్ బ్లాక్

కార్యాచరణ:

ఇన్‌పుట్‌లు

(ఎడమ వైపు)

డాటైన్ (7:0) —› ఇది పూర్తి ప్యాకేజీ కంటెంట్ డెస్క్ వద్దకు చేరుకుంటుంది — మూలం నుండి నేరుగా 8 బిట్స్ డేటా.

క్లిక్ చేయండి —+ ది కార్యాలయం గడియారం — ప్రతి టిక్ అంటే ఒక కొత్త అవకాశం కు లాగ్ లేదా పాస్ వెంట సమాచారం. detect_add FSM నుండి "కొత్త షిప్‌మెంట్ స్పాట్ చేయబడింది" అనే హెచ్చరిక, రిజిస్టర్‌ని కొత్త ప్యాకెట్ కోసం సిద్ధం చేయమని చెబుతుంది .

ఫైఫో_ఫుల్ / పూర్తి_రాష్ట్రం / లాఫ్_స్టేట్ —+ ఇవి ఉన్నాయి స్థితి నివేదికలు గురించి ది నిల్వ గదులు (FIFO లు) నిండిన , దాదాపు పూర్తి, లేదా పూర్తిగా అందుబాటులో ఉంది.

ld_స్టేట్ / lfd_స్టేట్ —+ డేటాను లోడ్ చేయడానికి లేదా ప్యాకెట్ యొక్క మొదటి భాగాన్ని లోడ్ చేయడానికి

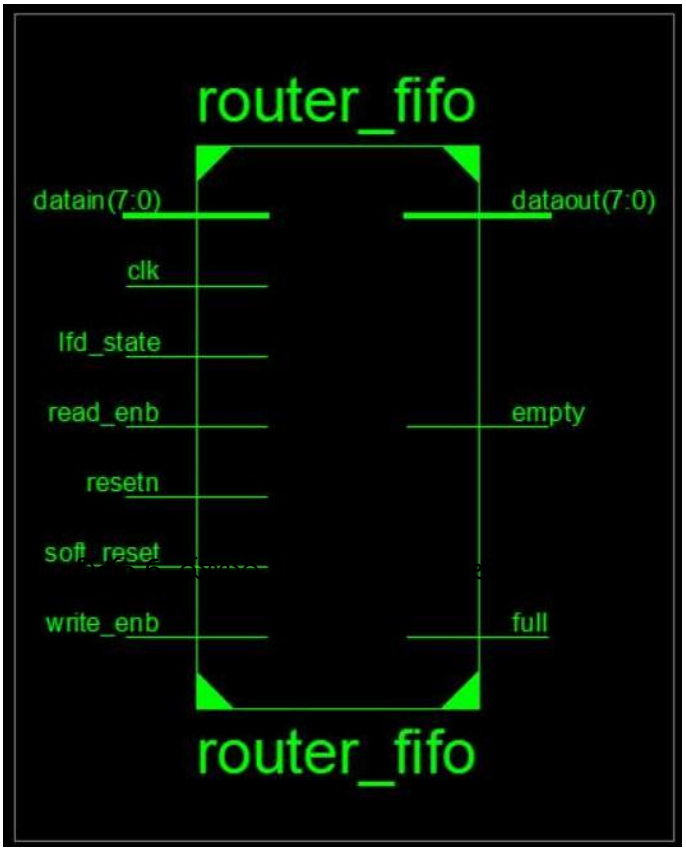
The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No 11, Nov 2025

FSM నుండి సూచనలు .
ప్యాకెట్_వాలిడ్ నిర్ధారిస్తుంది అది ది ఇన్‌కమింగ్ ప్యాకేజీ ఉంది చట్టబద్ధమైన మరియు నిల్వ చేయడానికి విలువైనది .

తిరిగి అమర్చబడిన / మొదటి_ఇంట్_రెగ్ — ది స్పష్టమైన డెస్క్ ఆదేశాలు — వారు తుడవండి అన్నీ తాత్కాలికమైన గమనికలు కాబట్టి తదుపరి షిప్‌మెంట్ కోసం రిజిస్టర్ తాజాగా ఉంది.
అవుట్‌పుట్‌లు (కుడి వైపు)
డౌట్ (7:0) —› ది జాగ్రత్తగా కాపీ చేయబడింది డేటా ప్యాకెట్, సిద్ధంగా ఉంది కోసం ది తరువాత విభాగం (ఫిఫో).
తప్పు —› అ ఎరుపు జెండా చెప్పడం అందరూ ఉంది ఉంది ఒక సమస్య తో ది రవాణా (ఇలా ఒక విఫలమైంది సమానత్వం
తనిఖీ చేయండి).
తక్కువ acket_valid —+ అ ముందస్తు హెచ్చరిక మనం దగ్గరగా ఉన్నామని ముగింపు యొక్క ఒక రవాణా.
పారిటీ_డన్ —› ఎ స్టాంపు యొక్క ఆమోదం చెప్పడం ది ప్యాకేజీ గడిచిపోయింది దాని సమగ్రత తనిఖీ.

రూటర్ FIFO తెలుగు in లో బ్లాక్
ది రూటర్_ఫిఫో ఉంది ఇష్టం ఒక తాత్కాలికమైన నిల్వ గది లో మా డిజిటల్ పోస్ట్ కార్యాలయం.
ఒకసారి రిసెప్షనిస్ట్ (రూటర్ reg) ధృవీకరించబడిన ప్యాకేజీ, FIFO దుకాణాలు అది సురక్షితంగా వరకు డెలివరీ ట్రక్ (అవుట్పుట్ పోర్ట్) సిద్ధంగా ఉంది కు తీసుకోండి దాన్ని దూరంగా. ఇది పనిచేస్తుంది లో ముందుగా వచ్చినవారు, ముందుగా అందించబడిన ఆర్డర్ — కేవలం ఇష్టం ఒక క్యూ వద్ద ఒక
టికెట్ కౌంటర్.

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No 11, Nov 2025



ఇన్‌పుట్‌లు (ఎడమ వైపు)

డాటైన్ (7:0) —› ఇది ఉంది ది ప్యాకేజీ ఉండటం ఉంచబడింది పై నిల్వ షెల్ఫ్.

క్లిక్ చేయండి —+ ది గిడ్డంగి గడియారం — ప్రతి టిక్ ఇస్తుంది ఒక అవకాశం కు స్టోర్ లేదా పంపండి బయటకు ఒక అంశం.

lfd_స్టేట్ —› అ సిగ్నల్ మాట్లాడుతూ, “ఇది ఉంది ది చాలా ముందుగా భాగం యొక్క ఒక కొత్త రవాణా — హ్యాండిల్ దానిని జాగ్రత్తగా."

read_enb ది డెలివరీ అభ్యర్థన — సమయం కు తీసుకోండి ది తరువాత ప్యాకేజీ ఆఫ్ ది షెల్ఫ్ మరియు పంపండి అది బయటకు. తిరిగి అమర్చబడింది / సాఫ్ట్_రీసెట్ —+ ది అత్యవసర పరిస్థితి స్పష్టమైన ఆదేశాలు — తక్షణమే ఖాళీ ది షెల్ఫ్ మరియు కొత్తగా ప్రారంభించండి .

write_enb ద్వారా —r ది అనుమతి జారిపోవు నుండి ఎఫ్ఎస్ఎం మాట్లాడుతూ, “అవును, స్టోర్ ఇది ప్యాకేజీ ఇప్పుడు."

అవుట్‌పుట్ (కుడి వైపు)

డేటాఅవుట్ (7:0) — ది ప్యాకేజీ ఉండటం చేతికిచ్చిన పైగా కోసం డెలివరీ.

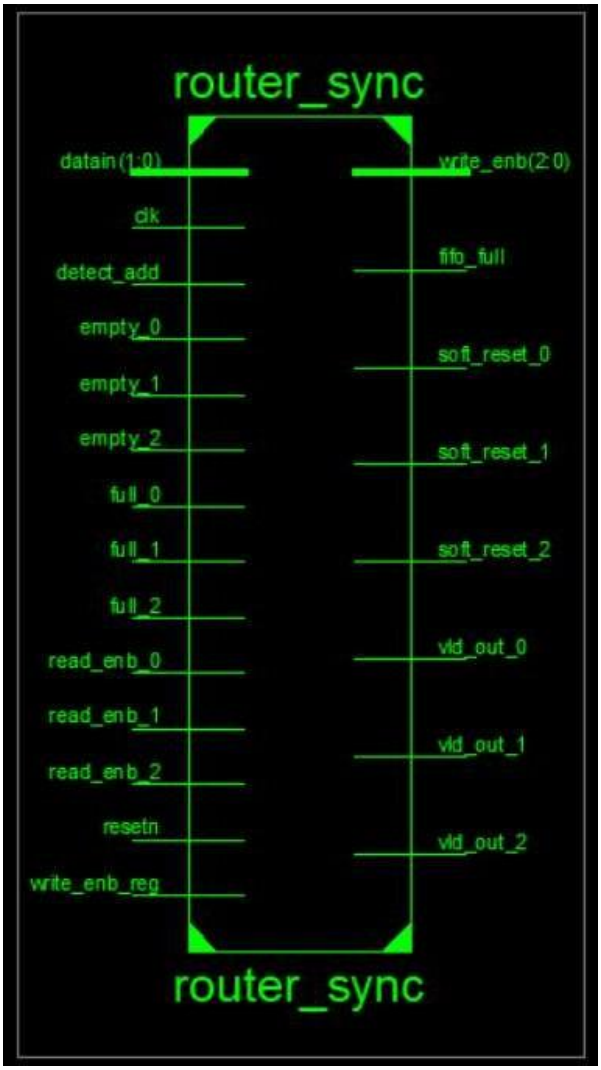
The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No 11, Nov 2025

ఖాళీ —+ అ గుర్తు మాట్లాడుతూ, “ఏమీ లేదు ఎడమ ఇక్కడ — అన్నీ ప్యాకేజీలు ఉన్నాయి పోయింది."
పూర్తి A హెచ్చరిక సంకేతం, “లేదు మరిన్ని స్థలం — ఆపండి పంపుతోంది ప్యాకేజీలు వరకు మేము తయారు చేయు గది."

రూటర్ సమకాలీకరణ బ్లాక్

రౌటర్ నెట్‌వర్క్-ఆన్-చిప్ (NoC) రౌటర్‌లోని డేటా ప్యాకెట్‌ల కోసం సింక్ బ్లాక్ ఎయిర్ ట్రాఫిక్ కంట్రోలర్ లాగా పనిచేస్తుంది. కంట్రోలర్ ప్లేన్‌లను సరైన టెర్మినల్‌లకు డైరెక్ట్ చేసినట్లే, రూటర్ సింక్ ఇన్‌కమింగ్ ప్యాకెట్‌లను వాటి గమ్యస్థాన చిరునామా ఆధారంగా సరైన అవుట్‌పుట్ FIFO కి డైరెక్ట్ చేస్తుంది. ఇది ప్రతి FIFO ను నిశితంగా గమనిస్తూ, ఏదీ ఎక్కువగా నింపబడలేదని నిర్ధారిస్తుంది మరియు చెల్లుబాటు అయ్యే డేటా డెలివరీకి సిద్ధంగా ఉన్నప్పుడు సిగ్నల్ ఇస్తుంది. ఒక టెర్మినల్ (FIFO) చాలా రద్దీగా మారుతుంది లేదా అవసరాలు క్లియరింగ్, అది ట్రిగ్గర్ చేయగలదు a కేవలం రీసెట్ చేయండి అది టెర్మినల్ లేకుండా కలవరపెట్టే ది ఇతరులు. ద్వారా నిర్వహణ ప్యాకెట్ ప్రవాహం లో ఇది మార్గం, రౌటర్ సమకాలీకరణ నిర్ధారిస్తుంది మృదువైన, సురక్షితమైన, మరియు సమర్థవంతమైన డేటా కదలిక ద్వారా ది రౌటర్.

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No 11, Nov 2025



చిత్రం 7. రూటర్ సమకాలీకరణ బ్లాక్

ఇన్‌పుట్‌లు (ఎడమ **వైపు)**

డేటాఇన్ [1:0] —+ ది చిరునామా ట్యాగ్ ఆన్ ది లేఖ చెప్పడం నువ్వు ఏది బుట్ట అది చెందినది కు (ఉదా., బుట్ట 0, 1, లేదా 2).

క్లిక్ చేయండి —+ ది టిక్ టిక్ చేయడం యొక్క ది కార్యాలయం గడియారం — నువ్వు పని లో సమకాలీకరణ తో ఇది. డిటెక్ట్_యాడ్ —+ మీరు ఒక లేఖను తీసుకొని చిరునామాను చూసిన క్షణం. empty_0 / ఖాళీ_1 / ఖాళీ_2 —+ బుట్ట 0 అయినా , 1, లేదా 2 ఉంది ఖాళీ. **పూర్తి_0 / పూర్తి_1 / f ul l _2** —+ బుట్టలో

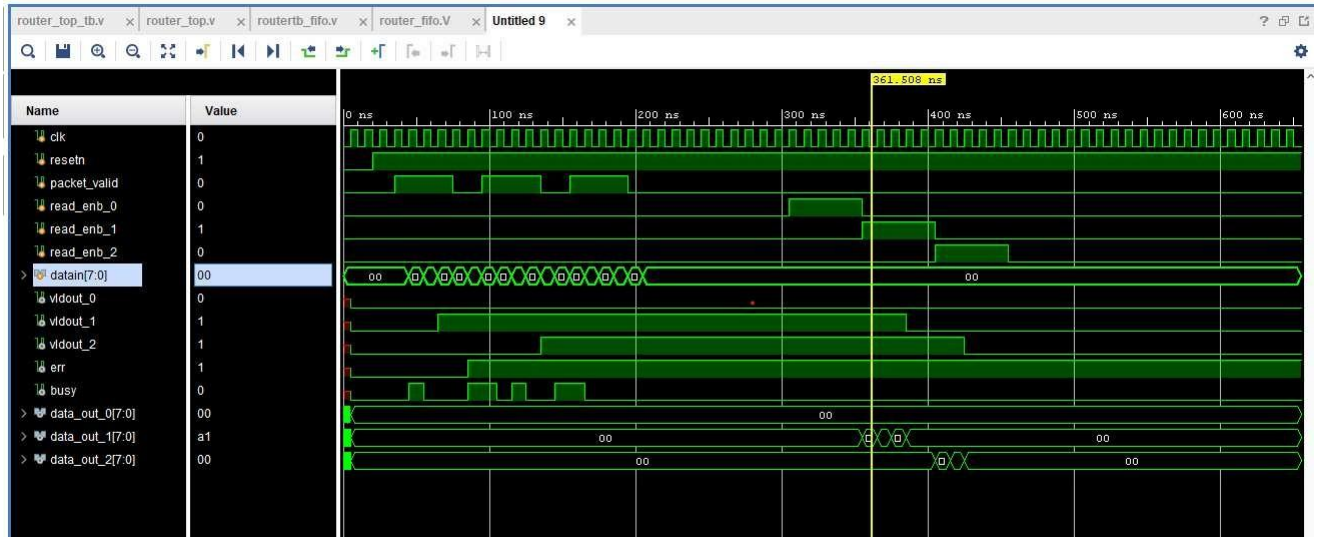
The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No 11, Nov 2025

ఉన్నాయా లేదా 0, 1, లేదా 2 ఉంది ఇప్పటికే నిండింది.
రీడ్_ఎన్బి_0 / చదవడం_enb_1 / చదవడం_enb_2 —+ డెలివరీ అని అడుగుతున్న వ్యక్తి వారు చెయ్యవచ్చు ఉత్తరాలు తీసుకోండి బుట్ట 0, 1, లేదా 2 నుండి .

తిరిగి అమర్చబడిన — పెద్దది "స్పష్టం అన్నీ బుట్టలు" బటన్
(క్రియాశీలత తక్కువ). write_enb_reg —› మీ సంసిద్ధత కు స్థలం
అక్షరాలు లో బుట్టలు.

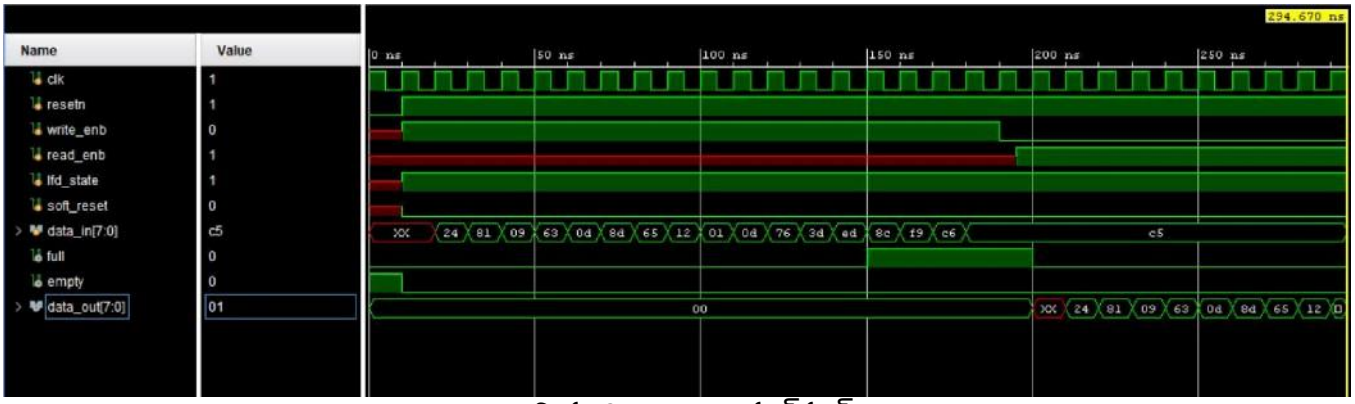
అవుట్‌పుట్‌లు (కుడి వైపు)
write_ enb [2:0] — సంకేతాలు కు నిజానికి స్థలం అక్షరాలు లోకి బుట్ట 0, 1, లేదా 2.
fifo_full నువ్వు అరవడం "ఆపు! ఇక ఉత్తరాలు లేవు!" ఎప్పుడు అన్నీ బుట్టలు ఉన్నాయి నిండింది.
సాఫ్ట్_రీసెట్_0 / సాఫ్ట్_రీసెట్_1 / సాఫ్ట్_రీసెట్_2 —+ ఆదేశం స్పష్టమైన బుట్ట 0, 1, లేదా 2
వ్యక్తిగతంగా. vld_out_0 / ద్వారా vilket_1 / vld_out_2 మీరు చెప్పు డెలివరీ వ్యక్తి "అవును, బుట్ట స ఉంది మీ కోసం చెల్లుబాటు అయ్యే మెయిల్ సిద్ధంగా ఉంది."

ప్రవర్తనాపరమైన **అనుకరణ** ఫలితాలు



చిత్రం .8. రూటర్ అవుట్‌పుట్

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No 11, Nov 2025

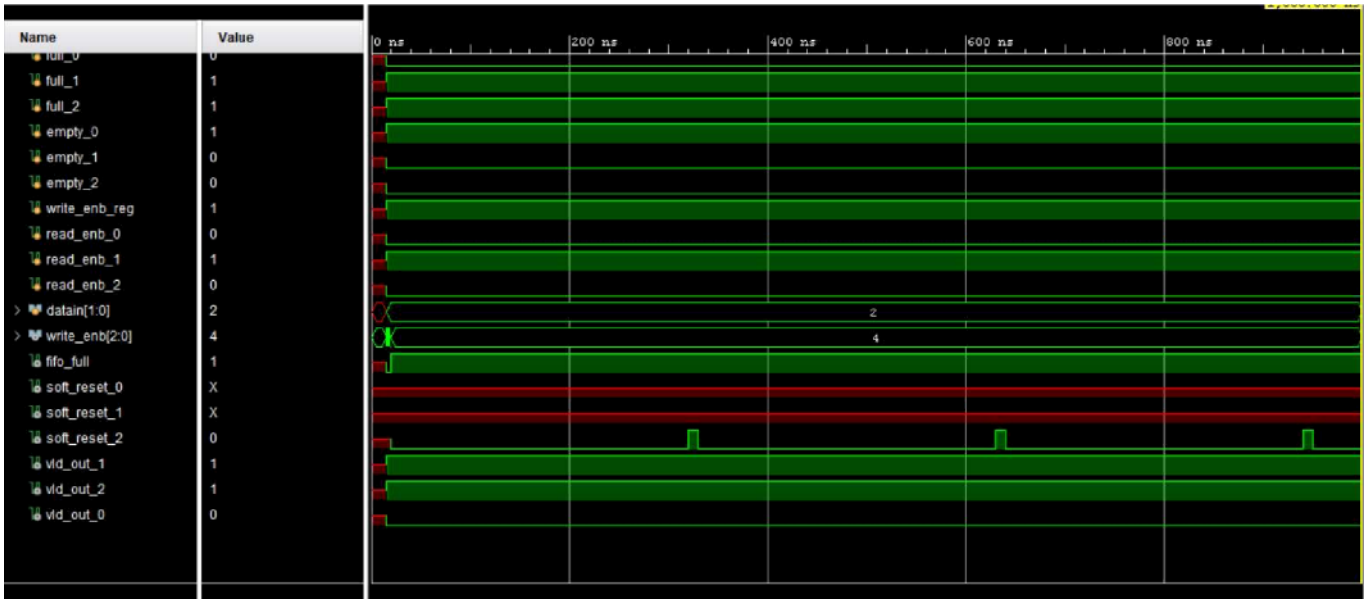


చిత్రం 9. FIFO అవుట్‌పుట్

చిత్రం 10.FSMఅవుట్‌పుట్

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No 11, Nov 2025

చిత్రం 11.r e g అవుట్పుట్



చిత్రం 12. సమకాలీకరణ అవుట్పుట్

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No 11, Nov 2025

ముగింపు

లో ఇది ప్రాజెక్ట్, మేము సెట్ బయటకు డిజైన్ ఎ అంతర్నిర్మిత బఫరింగ్‌తో VLSI రౌటర్ ఆన్-చిప్ డేటా బదిలీ చేయండి వేగంగా, మృదువైన, మరియు మరిన్ని నమ్మదగినది. ద్వారా జోడించడం FIFO తెలుగు in లో బఫర్‌లు మరియు తెలివైన రూటింగ్ తర్కం, ఈ వ్యవస్థ డేటాను వదలకుండా ఆకస్మిక ట్రాఫిక్ పేలుళ్లను నిర్వహించగలదు, ఆలస్యాన్ని తగ్గించగలదు మరియు ఒకే మార్గం కోసం బహుళ ఛానెల్‌లు పోటీ పడుతున్నప్పుడు కూడా సమాచారాన్ని ప్రవహించకుండా ఉంచగలదు. డిజైన్‌ను కనీస శక్తి మరియు చిప్ ప్రాంతాన్ని ఉపయోగించుకునేలా ట్యూన్ చేశారు, అదే సమయంలో అధిక నిర్గమాంశను అందిస్తారు, ఇది ఆచరణాత్మక ఎంపికగా మారింది కోసం ఆధునిక SoC లు మరియు డేటా-ఇంటెన్సివ్ వ్యవస్థలు. పరీక్షిస్తోంది మరియు FPGA తెలుగు in లో ఈ బఫర్డ్ రౌటర్ వేగం మరియు సామర్థ్యం రెండింటిలోనూ సాంప్రదాయ అన్‌బఫర్డ్ డిజైన్‌లను అధిగమిస్తుందని అమలు చూపించింది . సంక్షిప్తంగా, ఇది బిజీగా ఉన్న డిజిటల్ హైవేకి అదనపు లేన్‌లను మరియు మెరుగైన ట్రాఫిక్ నియంత్రణను ఇవ్వడం లాంటిది - ఇది విషయాలు కదులుతున్న, సరి సమయంలో తొందర గంట. చూస్తున్నాను ముందుకు, ఇది విధానం చేయగలిగి స్మార్ట్ పవర్ మేనేజ్‌మెంట్, ఫాల్ట్ టాలరెన్స్ మరియు ఇంకా ఎక్కువ డేటా రేట్లకు మద్దతుతో విస్తరించబడుతుంది, పేవింగ్ ది మార్గం కోసం తరువాతి తరం చిప్-టు-చిప్ కమ్యూనికేషన్.

ప్రస్తావనలు

[1]. రాబే , జెఎం, చంద్రకాసన్ , ఎ., & నికోలిక్ , బి. (2003). డిజిటల్ ఇంటిగ్రేటెడ్ సర్క్యూట్లు: ఒక డిజైన్ దృక్పథం (2వ ఎడిషన్ .). ప్రెంటిస్ హాల్. ఇది పుస్తకం అందిస్తుంది లోతైన అవగాహన డిజిటల్ సర్క్యూట్ డిజైన్, సహా □□□□ రౌటర్ ఆర్కిటెక్చర్లు, బఫర్ డిజైన్ మరియు ఆప్టిమైజేషన్ పద్ధతులు శక్తి మరియు ప్రాంత సామర్థ్యం కోసం

[2]. హెన్నెస్సీ, □□, □ ప్యాటర్సన్, □□ (2011). కంప్యూటర్ ఆర్కిటెక్చర్: ఎ క్వాంటిటేటివ్ అప్రోచ్ (5వ ఎడిషన్ .). మోర్గాన్ కౌఫ్మాన్ ప్రచురణకర్తలు. ఓ ఈ సూచన ఆర్కిటెక్చర్‌ను కవర్ చేస్తుంది ఆధునిక రౌటర్లు మరియు మారడం అంశాలు, సమర్పణ ఉపయోగకరమైన నేపథ్యం అర్థం చేసుకోవడానికి □□□□ సిస్టమ్స్‌లో రూటింగ్ మరియు నెట్‌వర్క్-ఆన్-చిప్ (□□□) డిజైన్.

[3]. యూ , హెచ్‌జె, లీ, కె., & కిమ్, జె. కె. (2008). తక్కువ శక్తి ఎన్.ఓ.సి. కోసం అధిక ప్రదర్శన SoC డిజైన్. సిఆర్‌సి నొక్కండి. □ ఈ పుస్తకం నెట్‌వర్క్-ఆన్-చిప్ (□□□) రౌటర్ల రూపకల్పనపై వివరణాత్మక అంతర్దృష్టులను అందిస్తుంది , ఇవి పవర్ మరియు స్పీడ్ ఆప్టిమైజేషన్‌పై దృష్టి సారించి □□□□ రౌటర్ డిజైన్‌కు నేరుగా వర్తిస్తాయి.

[4]. సాదిక్ , □□, □ కమర్ , □. (2017). "ఎ హై-స్పీడ్" బఫరింగ్ టెక్నిక్ "నెట్‌వర్క్-ఆన్-చిప్ ఆర్కిటెక్చర్‌ల కోసం." □□□□ ట్రాన్సాక్షన్స్ ఆన్ వెరీ లార్జ్-స్కేల్ ఇంటిగ్రేషన్ (□□□□) సిస్టమ్స్, 25(3), 1001-1011. □ ఈ పత్రం □□□ రౌటర్‌లలోని వివిధ బఫరింగ్ పద్ధతులను అన్వేషిస్తుంది మరియు థ్రూపుట్‌ను మెరుగుపరచడానికి మరియు తగ్గించడానికి పద్ధతులను అందిస్తుంది . జాప్యం, ఇది నేరుగా □□□□ రౌటర్‌ను మెరుగుపరుస్తుంది డిజైన్లు.

[5]. కౌల్ , ఎం., & దత్తా , పి. (2013). విఎల్ఎస్ఐ రూపకల్పన సిద్ధాంతం మరియు సాధన చేయండి. టాటా మెక్గ్రాహిల్ . ఓ అ □□□□ డిజైన్ సూత్రాలపై సమగ్ర వనరుతో, ఈ టెక్స్ట్ రౌటర్ డిజైన్ యొక్క ఆచరణాత్మక అంశాలను కవర్ చేస్తుంది, వేగవంతమైన డేటా బదిలీ కోసం బఫరింగ్ వ్యూహాలతో సహా.

[6]. వోల్ఫ్, డి., & గ్రీన్, డి. (2004). "డేటా నెట్వర్క్ల కోసం హై-పెర్ఫార్మెన్స్ రూటర్ల రూపకల్పన." □□□□ జర్నల్ ఆన్ సెలెక్టెడ్ ఏరియాస్ ఇన్ కమ్యూనికేషన్స్, 22(8), 1373- 1384. □ ఈ జర్నల్ వ్యాసం హై-స్పీడ్ నెట్వర్క్ల కోసం రౌటర్ల రూపకల్పనను చర్చిస్తుంది, రద్దీని తగ్గించడానికి మరియు డేటా ప్రవాహాన్ని ఆప్టిమైజ్ చేయడానికి బఫర్ నిర్వహణ పద్ధతులతో సహా .

[7]. వాంగ్, ఎస్., & లి, జెడ్. (2011). "□□□□ రూటర్ల కోసం సమర్థవంతమైన బఫర్ నిర్వహణ పథకం రూపకల్పన." □□□□ ట్రాన్సాక్షన్స్ ఆన్ కంప్యూటర్స్, 60(7), 1009-1018. □ ఈ పరిశోధన పత్రం □□□□ రౌటర్ల కోసం వినూత్న బఫర్ నిర్వహణ పథకాలను అందిస్తుంది, నిర్గమాంశ, విద్యుత్ వినియోగాన్ని సమతుల్యం చేయడానికి పద్ధతులను అందిస్తుంది, మరియు ఆలస్యం.

[8]. వంగల్ , ఎస్., మరియు ఇతరులు a1. (2007). "ఫ్లైఓవర్ లింక్లతో కూడిన 16-టైల్ 2□ మెష్ నెట్వర్క్-ఆన్-చిప్." □□□□ ఇంటర్నేషనల్ సాలిడ్-స్టేట్ సర్క్యూట్స్ కాన్ఫరెన్స్ (□□□□□), 2007, 124- 126. □ ఈ పత్రం □□□ రౌటర్ ఆర్కిటెక్చర్ మరియు పెద్ద వ్యవస్థలో అధిక-పనితీరు గల డేటా బదిలీ కోసం బఫర్ల ఏకీకరణ యొక్క ఆచరణాత్మక ఉదాహరణను అందిస్తుంది .

[9]. అగర్వాల్ , ఎం., & అగర్వాల్ , ఎ. (2014). సిస్టమ్ ఆన్ చిప్: ఆర్కిటెక్చర్ మరియు డిజైన్ పద్ధతులు. స్ప్రింగర్. □ బఫర్లు, పనితీరు ఆప్టిమైజేషన్ మరియు పెద్ద వ్యవస్థలలో ఏకీకరణపై ప్రత్యేక శ్రద్ధతో SoC అప్లికేషన్ల కోసం సమర్థవంతమైన రౌటర్లను రూపొందించడంలో అంతర్దృష్టులను అందిస్తుంది .

[10] . సైటో , ఎం., & తనకా, ఎస్. (2008). "డిజైన్ అండ్ ఆప్టిమైజేషన్ □□□□ ఆధారిత నెట్వర్క్-ఆన్-చిప్ కోసం రౌటర్." □□□□ లావాదేవీలు ఆన్ చాలా లార్జ్-స్కేల్ ఇంటిగ్రేషన్ (VLSI) వ్యవస్థలు, 16(11), 1469-1480. □ ఈ పత్రం ఒక ఉదాహరణ యొక్క రౌటర్ డిజైన్ కోసం విఎల్ఎస్ఐ వ్యవస్థలు, దృష్టి పెట్టడం ఆన్ బఫర్ వినియోగం మరియు వేగం ఆప్టిమైజేషన్ కోసం వేగవంతమైన డేటా బదిలీ.

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No11, Nov 2025
PP 201-219

Design and Implementation of a Buffer-Based High-Speed Data Transfer Router Using HDL

Nenavath Tharun[1] , Dr. Bommidi Sridhar[2], Banothu Bapuji [3],
Salivoju Laharika[4]
[1,4]*UG Student, Department of ECE, Scient Institute of technology, Ibrahimpatnam, Telangana, India,*
[2]*Department of ECE, Jawaharlal Nehru Technological University Hyderabad, Telangana, India.*
[3] *M. Tech, Dept of ECE, CVR college of Engineering, Ibrahimpatnam, Telangana, India*

Corresponding Author *: nenavaththarunnayak99@gmail.com

Abstract:
This project presents the design and implementation of a buffer-based high-speed data transfer router using Hardware Description Language (HDL), aimed at improving throughput and reducing latency in Network-on-Chip (NoC) systems. The proposed router architecture incorporates FIFO buffers at each port, a round-robin arbitration mechanism for fair access control, and an overloaded Code Division Multiple Access (CDMA) crossbar switch for simultaneous multi-channel communication. The buffering mechanism efficiently manages bursty traffic and prevents data loss, while the CDMA-based switching enhances bandwidth utilization without significant area or power overhead. The design was described in Verilog HDL, simulated using Xilinx Vivado for functional verification, and synthesized for FPGA implementation on an Artix-7 device. Performance analysis demonstrated low latency, improved throughput, and efficient resource utilization compared to conventional unbuffered designs. This work provides a scalable and power-efficient router solution suitable for high-performance SoCs and advanced NoC architectures.

Keywords: VLSI Router, Network-on-Chip (NoC), Verilog HDL, Power Optimization, Performance, Area Efficiency, Data Communication, Xilinx Vivado, Zynq Platform, Packet Switching, SoC Design.

1. Introduction

In modern **VLSI** and **System-on-Chip (SoC)** architectures, fast and efficient communication between multiple **Processing Elements (PEs)** is critical to overall system performance. As the number of components on a chip continues to grow, traditional bus-based interconnect methods

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No11, Nov 2025
PP 201-219

have shown clear drawbacks—they are harder to scale, tend to create performance bottlenecks, and often consume more power than is acceptable in high-performance or power-sensitive designs.

To address these challenges, the **Network-on-Chip (NoC)** paradigm has emerged as a scalable and efficient alternative. In an NoC, **routers** and **links** replace the old shared bus, connecting system components in structured topologies such as mesh or torus. This approach allows multiple data transfers to happen at the same time, significantly improving throughput while reducing congestion compared to older designs.

At the heart of every NoC lies the **router**, which acts much like a traffic controller for on-chip data. Its primary job is to receive packets from one port and forward them to the correct destination port. This process involves several coordinated steps:

- **Packet Reception** – Data packets arrive at one of the router's input ports.
- **Buffering** – If the required output port is busy, the incoming packet is stored temporarily in a buffer.
- **Routing Computation** – The router decides where to send the packet next, using algorithms such as **deterministic XY routing** or adaptive routing methods.
- **Arbitration** – If more than one packet wants to use the same output port, the arbiter decides which one goes first, often using **round-robin** or priority-based methods.
- **Switching** – The router's internal crossbar connects the selected input to the output, enabling the packet to move forward.
- **Packet Transmission** – The packet is sent to the next router or its final destination.

While this seems straightforward, things become more complicated under **heavy traffic**. When many packets are competing for the same resources, delays and packet drops can occur. This is where **buffers** play an essential role.

Buffers act as temporary storage areas for packets that cannot be forwarded immediately. Their main functions include:

- **Handling Congestion** – Store packets when output ports are busy, preventing data loss.
- **Reducing Latency** – Keep data moving smoothly by enabling pipelining, where multiple packets are processed at different stages simultaneously.
- **Boosting Throughput** – Allow continuous data flow, even when traffic is bursty.
- **Supporting Flow Control** – Work with credit-based or handshake protocols to avoid buffer overflows and underflows.
- **Balancing Workloads** – Absorb sudden traffic spikes so that the router has time to process them efficiently.

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No11, Nov 2025
PP 201-219

In essence, routers ensure **where** data goes, while buffers ensure **how smoothly** it gets there. Together, they form the backbone of high-speed, reliable on-chip communication. Without buffers, even the fastest router would struggle under peak loads, much like a busy intersection without a holding lane for waiting cars.
By combining intelligent routing, fair arbitration, and well-managed buffering, modern NoC routers achieve the **low latency, high throughput, and reliability** demanded by today's complex VLSI systems.

2. Literature Survey

Kumar et al. (2002) were among the first to propose a structured **Network-on-Chip (NoC)** architecture, introducing scalable mesh-based topologies and packet-switched communication to overcome the bandwidth and scalability limitations of traditional bus-based systems. Their methodology set a benchmark for evaluating NoC performance in terms of latency, throughput, and power efficiency. Building on these foundations, Bjerregaard and Mahadevan (2006) presented a comprehensive survey of NoC design considerations, including topology selection, flow control, and routing strategies, stressing the need for balanced trade-offs between performance and hardware cost. In the area of on-chip switching, Bell et al. (2001) introduced **Code-Division Multiple Access (CDMA)** for multiprocessor interconnects, exploiting orthogonal spreading codes to enable concurrent communications over a shared medium, thereby reducing arbitration complexity and guaranteeing fixed latency. Nikolic et al. (2008) extended this concept by designing scalable CDMA peripheral buses that minimized pin counts and wiring complexity, improving communication between processing elements and peripherals. Lai et al. (2004) proposed the **CT-Bus** architecture, combining CDMA with Time-Division Multiple Access (TDMA) to enhance performance in heterogeneous traffic environments.

2. Proposed Architecture

The proposed VLSI router will be designed with the following architectural features:
1.VLSI Router with Buffer for Fast Data Transfer: This block shows a high-level view of a VLSI router that has an integrated buffer to help speed up data transfer between different directions — North, South, East, and West. You can think of it like a four-way traffic intersection for digital data. The router is in the middle, and the buffer acts like a holding bay where data

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No11, Nov 2025
PP 201-219

waits before being sent out. This buffering prevents congestion and ensures smooth flow, just like cars queuing in a small parking area before entering a busy road.

2. Router and Buffer Connection: Here, the diagram simplifies the concept by directly showing a buffer connected to a router. The buffer is essentially temporary storage for incoming or outgoing data packets, while the router decides where to send them next. In everyday terms, imagine the buffer as a small waiting lounge, and the router as the receptionist who directs people to the right office.

3. Input-Buffered Router: This diagram is a bit more detailed. Data enters from the North and is stored in a FIFO (First-In-First-Out) queue — like people lining up in order of arrival. The routing logic decides which output port each packet should take. Since multiple packets may want to go the same way, arbiters are used to decide who gets to go first. The arbiter works like a traffic light, ensuring fairness and avoiding collisions in data paths.

4. Router Network Topology: This grid of boxes represents multiple routers connected together, forming a network. Each "R" is a router, while some boxes are labeled "W" for West, "E" for East, and "S" for South — indicating directional positions in the network. It's similar to a city map where intersections (routers) connect roads (communication links) to help traffic (data) flow across the entire city (chip).

5. Local PE to Router Connection: This diagram shows how a local processing element (PE) — essentially the "brain" or computing core — connects to a router. The PE generates data that needs to travel to other parts of the chip, and the router is the gateway to send and receive that data. Think of the PE as a home, and the router as the local post office. All outgoing and incoming letters (data) go through it.

6. Intelligent Interconnect: Finally, the last diagram shows a more complex router with inputs from multiple directions — West, East, North, South, and Local. It's called "intelligent" because it can decide the most efficient path for data to travel, just like a smart GPS system that dynamically reroutes traffic based on conditions. It ensures the shortest travel time and prevents data congestion inside the chip network.

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No11, Nov 2025
PP 201-219

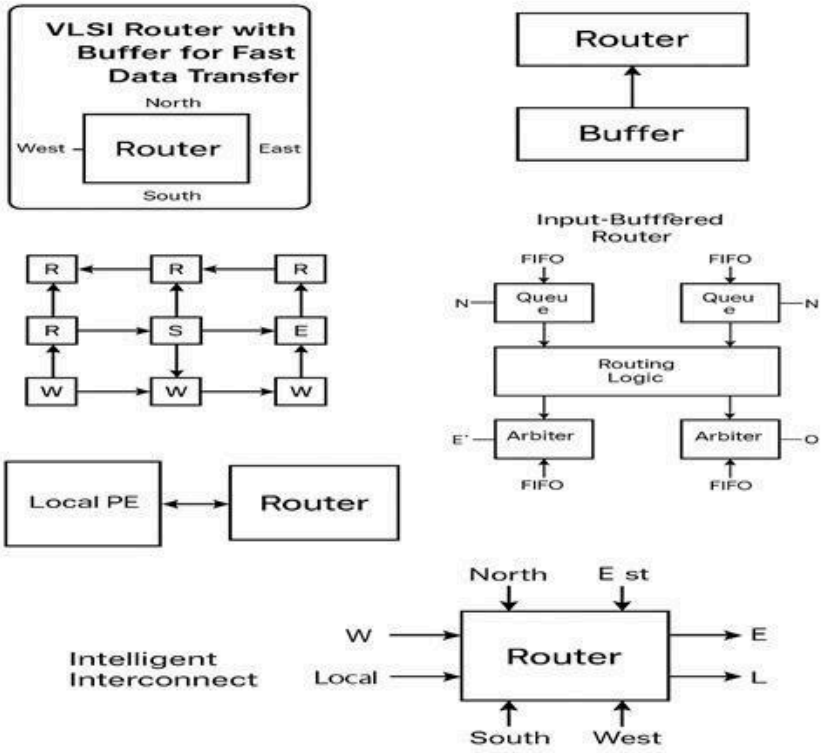
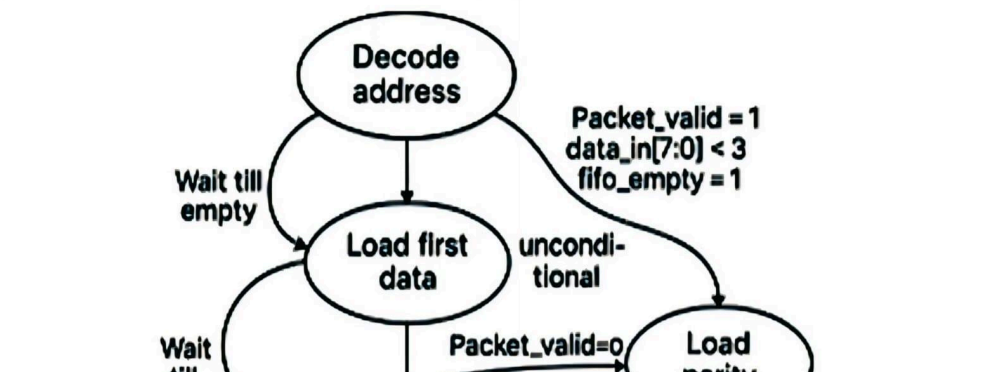


Fig.1. Proposed Architecture

4. Router FSM State Diagram

A Router FSM is like the traffic controller inside a smart post office. When a package (data packet) arrives, this controller decides where it should go, when it can be stored, and when it can be sent out — all while making sure no package gets lost or mixed up. It first reads the label (decode address) to figure out the correct delivery counter (output port). If there's room in the storage bin (FIFO), it places the package in line; if the bin is full, it waits until space frees up. Once the package is stored, it also checks the seal (parity check) to ensure nothing was damaged in transit. Just like a good post office manager keeps the mail flowing smoothly, the Router FSM keeps digital data moving orderly, on time, and error-free inside a network.



The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No11, Nov 2025
PP 201-219

Fig.2. Router FSM State Diagram

Functionality:

The functionality of this FSM for FIFO-based packet handling can be described step-by-step as follows:

1. Address Decoding: The FSM begins in the Decode Address state, where it checks the packet's destination address (data_in[7:0] < 3) and verifies that the packet is valid (Packet_valid = 1) and that the corresponding FIFO is empty. This ensures that incoming data is directed to the correct FIFO output channel.

2. Loading the First Data Byte: Moves to Load First Data, storing the first byte of the packet into the FIFO.Waits for FIFO readiness (empty state) before accepting more data.

3. Loading Remaining Data: In the Load Data state, subsequent bytes of the packet are loaded into the FIFO continuously. If the FIFO becomes full (fifo_full = 1), the FSM temporarily moves to the FIFO Full State and waits for space to become available.

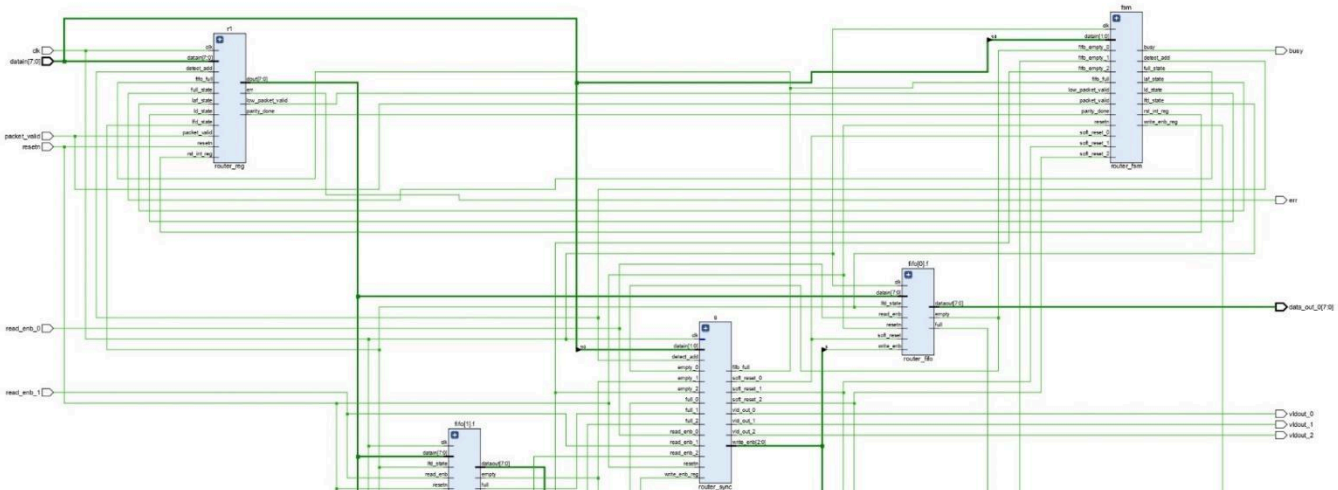
The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No11, Nov 2025
PP 201-219

4. Handling Full FIFO: In the FIFO Full State, the system pauses data writing until fifo_full becomes 0 again. Once there's space, it resumes loading the remaining data.
5. Loading Parity Byte: When all packet data is received (Packet_valid = 0), the FSM moves to Load Parity to store the parity byte. This parity byte will be used to check for transmission errors.
6. Parity Checking: The FSM enters the Check Parity Error state, comparing the calculated parity from the received data with the received parity byte. If they match, no error is reported; if they differ, an error signal is raised.
7. Returning to Idle: After parity verification, the FSM resets back to Decode Address to wait for the next packet.

5. Result

RTL Schematic

This design is like a well-organized postal sorting center, but for digital data. It's built from six main "departments": the FSM Router, the Router Register, the Sync Unit, and three FIFO buffers—one for each output path.
FSM Router (router_fsm) – Think of this as the traffic controller. It decides where each incoming data packet should go, generates the control signals, and keeps everything running in the right order. It tells the registers when to capture data and instructs each FIFO when to store or release it.
Router Register (router_reg) – This is the reception desk. Whenever a packet arrives, it stores the incoming status, data, and parity bits. But it only updates its records when the FSM gives the green light—making sure nothing gets overwritten by accident.
Three FIFO Buffers (router_fifo0, router_fifo1, router_fifo2) – These are like holding bays for outgoing mail. Each FIFO is linked to one output port and temporarily stores data until it's ready to be sent out. Which FIFO gets the data depends on the FSM's routing decision.
Sync Unit (router_sync) – This is the messenger that ensures smooth communication between the FSM and the FIFOs. It keeps all parts in sync so the single input port can send data cleanly and reliably to any of the three output ports without mix-ups.



The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No11, Nov 2025
PP 201-219

Fig.3. RTL Schematic diagram of a router

Router FSM Block

The router_fsm is like the chief traffic officer in a busy digital intersection. It doesn't actually carry the data itself — instead, it watches everything, makes decisions, and signals other blocks to take action at exactly the right time.

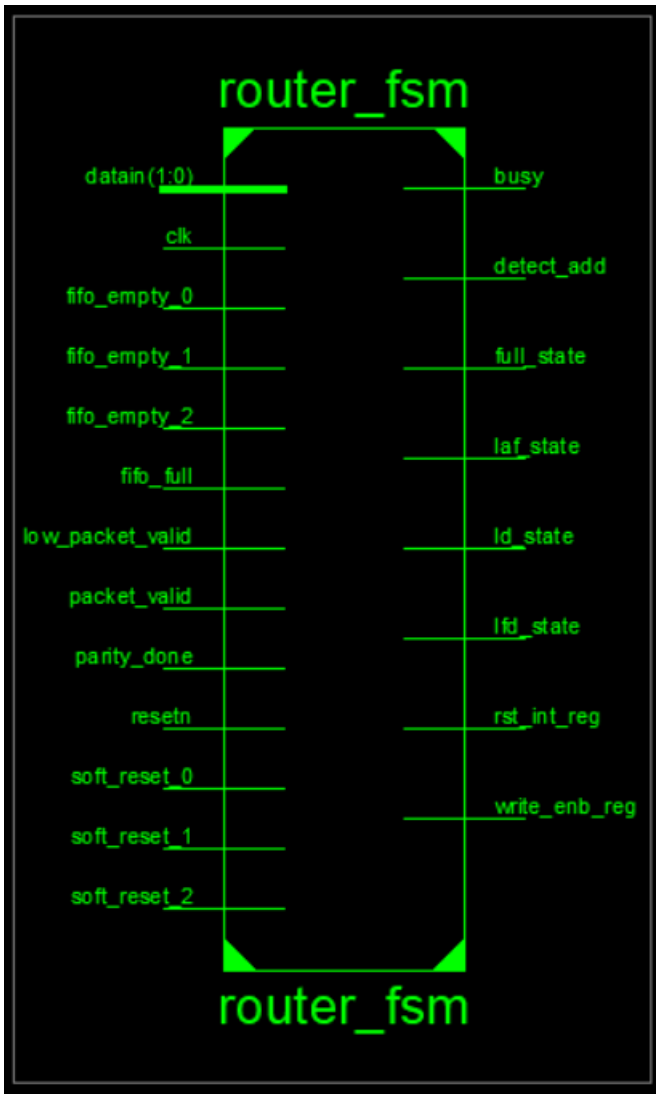


Fig.4.Router FSM Block

Functionality:

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No11, Nov 2025
PP 201-219

Inputs (Left Side)

datain(1:0) → These are like the destination tags on parcels. The FSM reads them to figure out which output port the data should be sent to.

fifo_empty_0 / fifo_empty_1 / fifo_empty_2 → These flags tell the FSM whether each output's storage bin (FIFO) is empty.

fifo_full → This is like a "warehouse is full" sign. If the FIFO is full, the FSM knows it needs to pause sending more data there.

low_packet_valid / packet_valid → These are like "new shipment has arrived" signals. They tell the FSM when incoming data is ready to process.

parity_done → This is a "quality check passed" flag. It tells the FSM the data's parity check is complete.

resetn / soft_reset_0/1/2 → These are the emergency stop or restart buttons for the FSM or for specific FIFOs.

clk → The heartbeat that keeps everything moving in sync.

Outputs (Right Side)

busy → A sign to the outside world saying, "Hold on, I'm working right now."

detect_add → The moment the FSM identifies the destination address of the packet.

full_state / laf_state → Internal states telling the system if storage is full or almost full.

ld_state / lfd_state → States for when the FSM should load data or load the first data word into a FIFO.

rst_int_reg → Orders to reset the internal register in the router.

write_enb_reg → The green light telling the router register, "Go ahead, capture this incoming data."

Router Register Block

The router_reg is like the reception desk in our digital post office.

When a new shipment of data arrives, this is the first stop. The receptionist (router_reg) carefully records all the details, keeps them safe, and only passes them forward when the manager (FSM) says it's the right time.

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No11, Nov 2025
PP 201-219

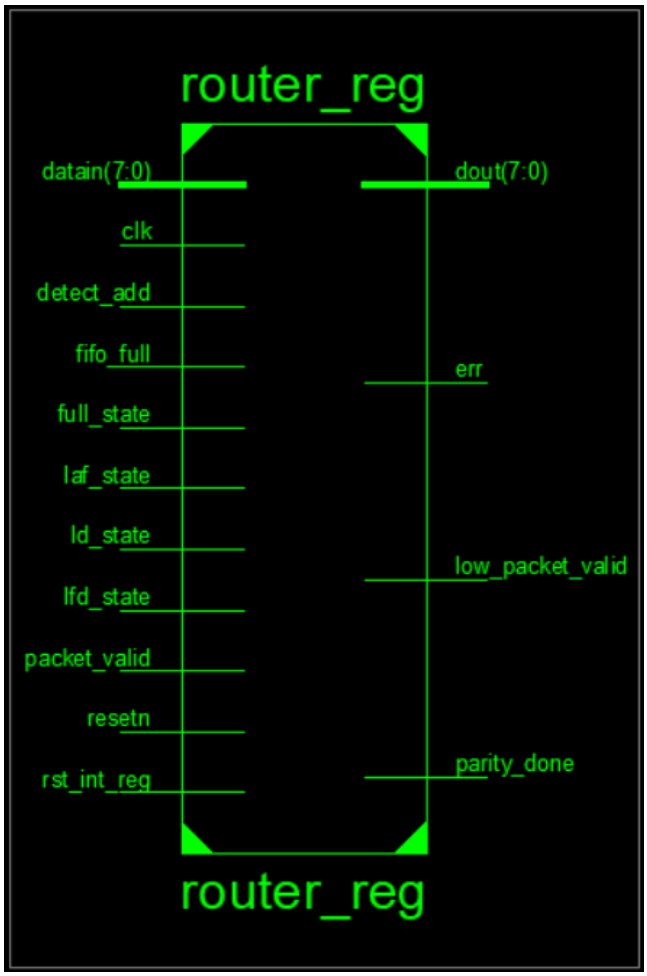


Fig.5.Router reg Block

Functionality:

Inputs (Left Side)

datain(7:0) → This is the full package content arriving at the desk — 8 bits of data straight from the source.

clk → The office clock — every tick means a new chance to log or pass along information.

detect_add → A "new shipment spotted" alert from the FSM, telling the register to prepare for a new packet.

fifo_full / full_state / laf_state → These are status reports about the storage rooms (FIFOs) — full, nearly full, or completely available.

ld_state / lfd_state → Instructions from the FSM to either load data or load the first piece of the packet.

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No11, Nov 2025
PP 201-219

packet_valid → Confirms that the incoming package is legitimate and worth storing.
resetn / rst_int_reg → The clear desk commands — they wipe all temporary notes so the register is fresh for the next shipment.
Outputs (Right Side)
dout(7:0) → The carefully copied data packet, ready for the next department (FIFO).
err → A red flag telling everyone there's been a problem with the shipment (like a failed parity check).
low_packet_valid → A heads-up that we're near the end of a shipment.
parity_done → A stamp of approval saying the package has passed its integrity check.

Router FIFO Block
The router_fifo is like a temporary storage room in our digital post office.
Once the receptionist (router_reg) hands over a verified package, the FIFO stores it safely until the delivery truck (output port) is ready to take it away. It works in a first-come, first-served order — just like a queue at a ticket counter.

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No11, Nov 2025
PP 201-219

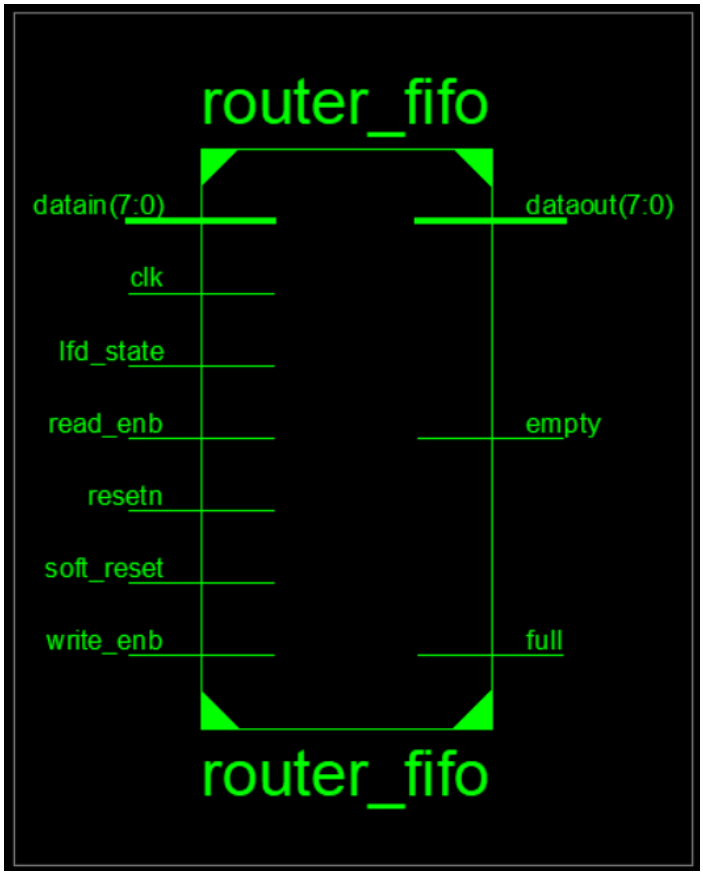


Fig.6.Router FIFO Block

Inputs (Left side)

datain(7:0) → This is the package being placed onto the storage shelf.

clk → The warehouse clock — each tick gives a chance to store or send out an item.

lfd_state → A signal saying, "This is the very first part of a new shipment — handle it carefully."

read_enb → The delivery request — time to take the next package off the shelf and send it out.

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No11, Nov 2025
PP 201-219

resetn / soft_reset → The emergency clear commands — instantly empty the shelf and start fresh.
write_enb → The permission slip from the FSM saying, "Yes, store this package now."

Output (Right Side)
dataout(7:0) → The package being handed over for delivery.
empty → A sign saying, "Nothing left here — all packages are gone."
full → A warning sign, "No more space — stop sending packages until we make room."

Router Synchronization Block

The router_sync block acts like an air traffic controller for data packets in a Network-on-Chip (NoC) router. Just as a controller directs planes to the correct terminals, router_sync directs incoming packets to the right output FIFO based on their destination address. It keeps a close watch on each FIFO, ensuring none are overfilled, and it signals when valid data is ready for delivery. If a terminal (FIFO) becomes too crowded or needs clearing, it can trigger a reset for just that terminal without disturbing the others. By managing packet flow in this way, router_sync ensures smooth, safe, and efficient data movement through the router.

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No11, Nov 2025
PP 201-219

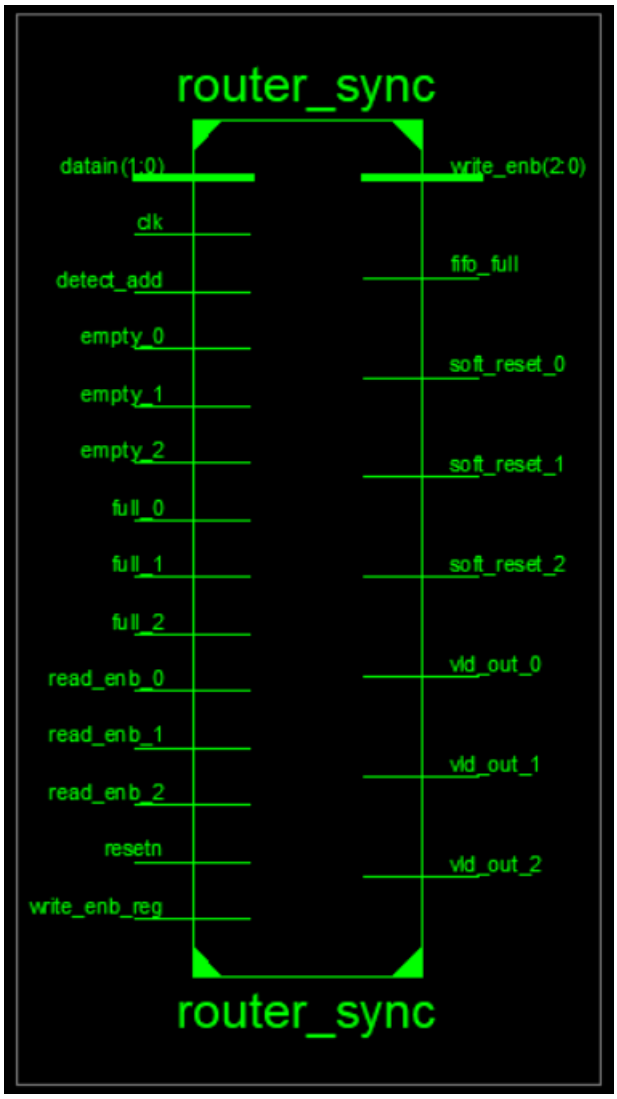


Fig.7.Router sync Block

Inputs (Left Side)

datain[1:0] → The address tag on the letter telling you which basket it belongs to (e.g., basket 0, 1, or 2).

clk → The ticking of the office clock – you work in sync with this.

detect_add → The moment you pick up a letter and look at the address.

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No11, Nov 2025
PP 201-219

empty_0 / empty_1 / empty_2 → Whether basket 0, 1, or 2 is empty.
full_0 / full_1 / full_2 → Whether basket 0, 1, or 2 is already full.
read_enb_0 / read_enb_1 / read_enb_2 → Delivery person asking if they can take letters from basket 0, 1, or 2.
resetn → Big "clear all baskets" button (active low).
write_enb_reg → Your readiness to place letters in baskets.

Outputs (Right Side)
write_enb[2:0] → Signals to actually place letters into basket 0, 1, or 2.
fifo_full → You shout "STOP! No more letters!" when all baskets are full.
soft_reset_0 / soft_reset_1 / soft_reset_2 → Command to clear basket 0, 1, or 2 individually.
vld_out_0 / vld_out_1 / vld_out_2 → You tell the delivery person "Yes, basket X has valid mail ready for you."

Behavioral Simulation Results

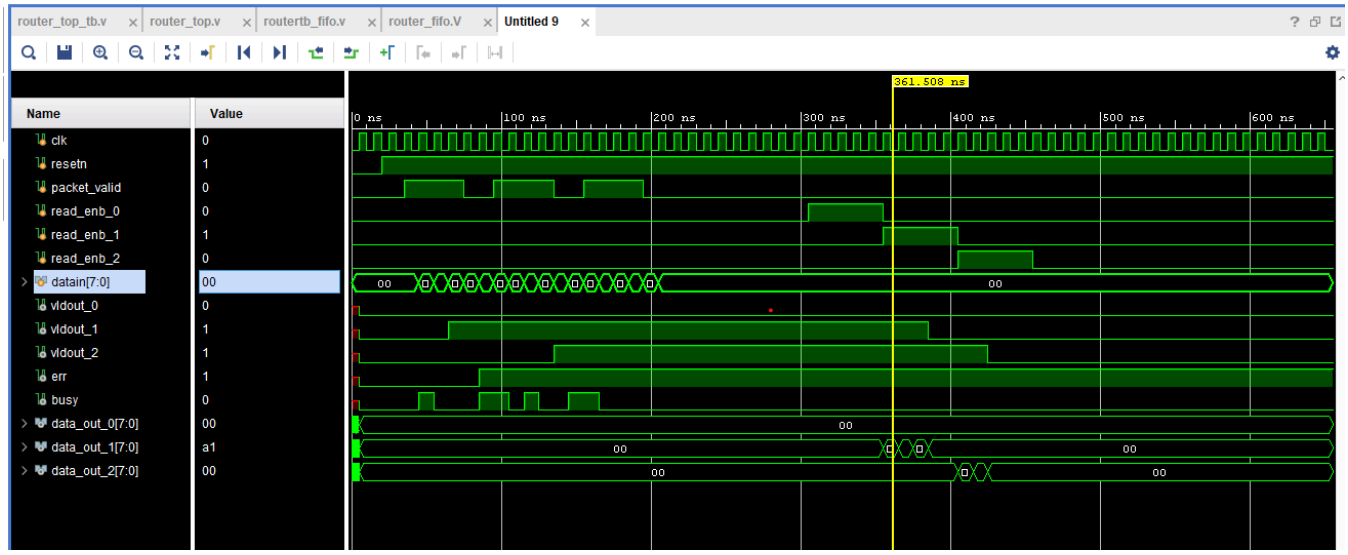


Fig.8. Router Output

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No11, Nov 2025
PP 201-219

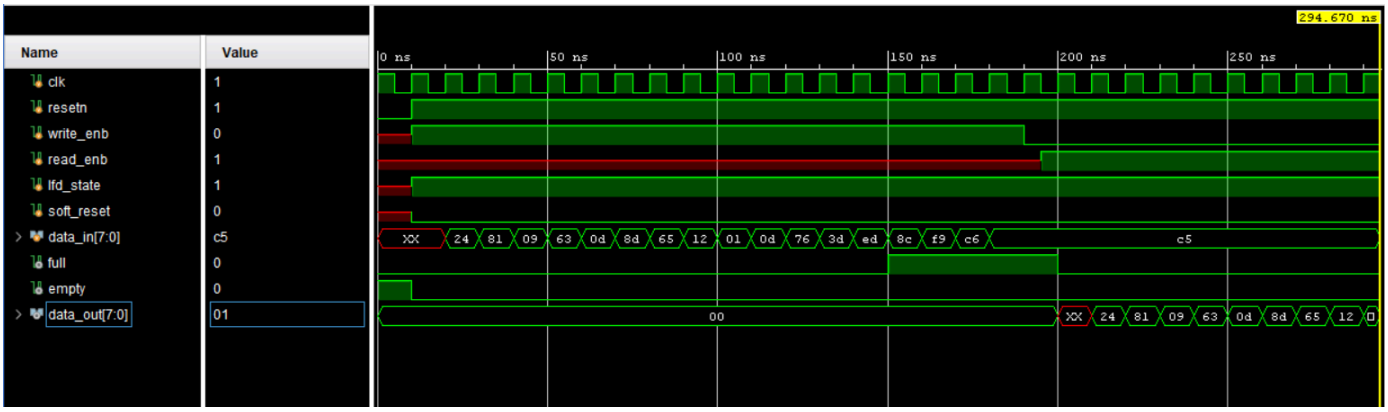


Fig.9.FIFO Output

Fig.10.FSM Output

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No11, Nov 2025
PP 201-219

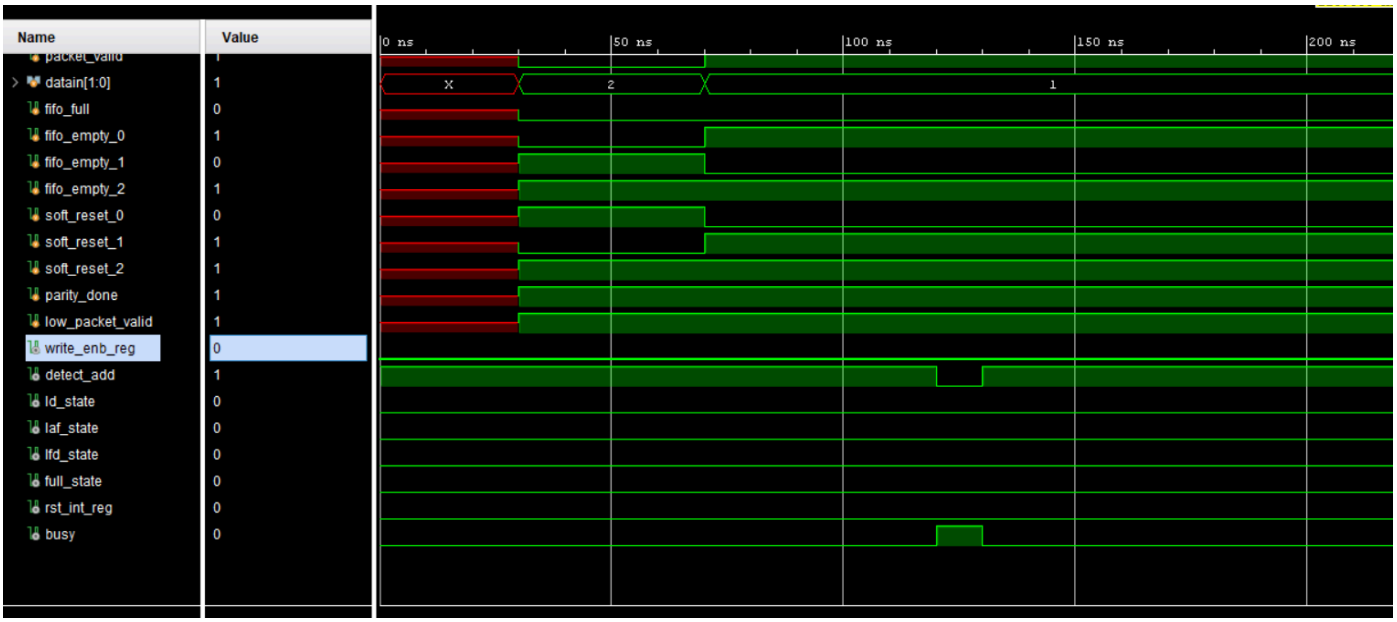


Fig.11.reg output

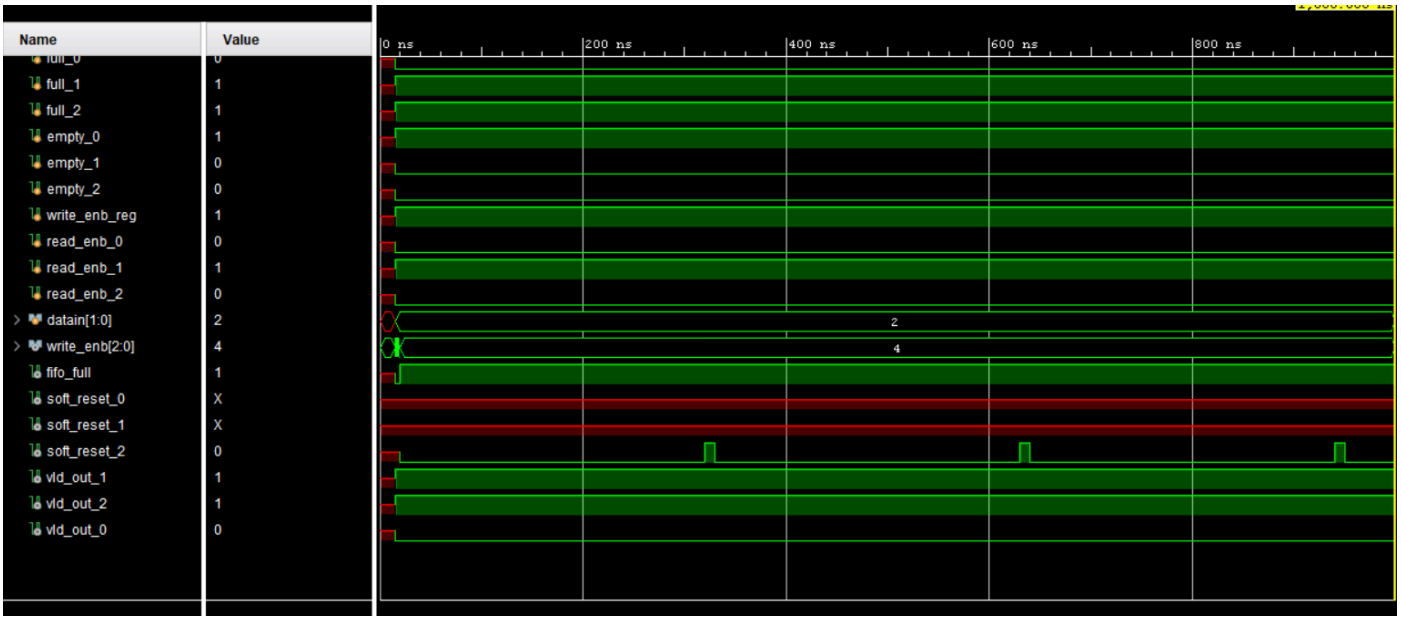


Fig.12.sync output

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No11, Nov 2025
PP 201-219

Conclusion

In this project, we set out to design a VLSI router with built-in buffering to make on-chip data transfer faster, smoother, and more reliable. By adding FIFO buffers and smart routing logic, the system can handle sudden bursts of traffic without dropping data, reduce delays, and keep information flowing even when multiple channels compete for the same path. The design was tuned to use minimal power and chip area while still delivering high throughput, making it a practical choice for modern SoCs and data-intensive systems. Testing and FPGA implementation showed that this buffered router outperforms traditional unbuffered designs in both speed and efficiency. In short, it's like giving a busy digital highway extra lanes and better traffic control—it keeps things moving, even during rush hour. Looking ahead, this approach could be extended with smarter power management, fault tolerance, and support for even higher data rates, paving the way for next-generation chip-to-chip communication.

References

[1]. Rabaey, J. M., Chandrakasan, A., & Nikolic, B. (2003). Digital Integrated Circuits: A Design Perspective (2nd ed.). Prentice Hall. ○ This book provides an in-depth understanding of digital circuit design, including VLSI router architectures, buffer design, and optimization techniques for power and area efficiency

[2]. Hennessy, J. L., & Patterson, D. A. (2011). Computer Architecture: A Quantitative Approach (5th ed.). Morgan Kaufmann Publishers. ○ This reference covers the architecture of modern routers and switching elements, offering useful background for understanding routing in VLSI systems and network-on-chip (NoC) design.

[3]. Yoo, H. J., Lee, K., & Kim, J. K. (2008). Low Power NoC for High Performance SoC Design. CRC Press. ○ This book provides detailed insights into the design of Network-on-Chip (NoC) routers, which are directly applicable to VLSI router design, with a focus on power and speed optimization.

[4]. Sadiq, M. T., & Qamar, F. (2017). "A High-Speed Buffering Technique for Network-on-Chip Architectures." IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, 25(3), 1001-1011. ○ This paper explores various buffering techniques in NoC routers and provides methods for improving throughput and reducing latency, which can directly enhance VLSI router designs.

[5]. Kaul, M., & Dutta, P. (2013). VLSI Design Theory and Practice. Tata McGrawHill. ○ A comprehensive resource on VLSI design principles, this text covers practical aspects of router design, including buffering strategies for fast data transfer.

[6]. Wolfe, D., & Green, D. (2004). "Design of High-Performance Routers for Data Networks." IEEE Journal on Selected Areas in Communications, 22(8), 1373- 1384. ○ This journal article discusses the design of routers for high-speed networks, including buffer management techniques for minimizing congestion and optimizing data flow.

[7]. Wang, S., & Li, Z. (2011). "Design of an Efficient Buffer Management Scheme for VLSI Routers." IEEE Transactions on Computers, 60(7), 1009-1018. ○ This research paper presents innovative buffer management schemes for VLSI routers, offering techniques to balance throughput, power consumption, and delay.

The Journal of Computational Science and Engineering (TJCSE)
ISSN 2583-9055 (Media Online)
Vol 3, No11, Nov 2025
PP 201-219

[8]. Vangal, S., et al. (2007). "A 16-Tile 2D Mesh Network-on-Chip with Flyover Links." IEEE International Solid-State Circuits Conference (ISSCC), 2007, 124- 126. ○ This paper presents a practical example of NoC router architecture and the integration of buffers for high-performance data transfer in a large system.

[9]. Agarwal, M., & Agarwal, A. (2014). System on Chip: Architecture and Design Methods. Springer. ○ Provides insights into designing efficient routers for SoC applications, with particular attention to buffers, performance optimization, and integration into larger systems.

[10]. Saito, M., & Tanaka, S. (2008). "Design and Optimization of a Router for VLSI Based Network-on-Chip." IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, 16(11), 1469-1480. ○ This paper provides an example of router design for VLSI systems, focusing on buffer utilization and speed optimization for fast data transfer.