RDMA Smart NIC (*RSNIC IP*)

May 2023 Status (January 17 - May 15)

Table of Contents

May 2023 **3**

Week 3 (5/15) 3

Week 2 (5/9) 3

Week 1 (5/2) 3

April 2023 **4**

Week 4 (04/25) 4

Week 3 (04/18) 5

Week 2 (04/11) 5

Week 1 (04/04) 6

March 2023 **8**

Week 4 (03/28) 8

Week 3 (03/21) 8

Week 2 (03/14) 9

Week 1 (03/07) 9

February 2023 **10**

Week 4 (02/28) 10

Week 3 (02/21) 10

Week 2 (02/14) 11

Week 1 (02/07) 12

January 2023 **13**

Week 4 (01/31) 13

Week 3 (01/24) 13

Week 2 (01/17) 14

May 2023

Week 3 (5/15)

1. Create function files srq.c and srq.h
2. create handle to mailbox command to create_srq context.
3. create handle to mailbox to query available srq.
4. Create a handle to the mailbox to destroy srq context after use.
5. Create a handle on qp to determine if the receive queue RQ is Srq or RQ.
6. Added firmware in mlx5_fq.c, mlx5_fq.h, Qp.c and Qp.h
7. Added firmware to initialized ring for uar idx 6 to 15.
8. Modify firmware on how the FPGA handles the new notification from the host.
9. Added firmware to get net wqe from 0x800 ring.
10. Modified qp req routine to add fetching of doorbell records to confirm if the host stopped from sending requests.
11. Added firmware to create packet from work request to packet being transmitted to QSFP.
12. Correct wqe_cnt to be passed cqe, use running wqe_cnt instead of masking on sq/rq size.

Week 2 (5/9)

1. In atomic operation, QP host is using compose of send queue has offset of 4WQE block. 0x40. Fetching offset 0 instead of 1.
2. Say they already finished one page and are going to loop back from the start. Going back to fetch offset 0 and not offset 1. Add offset 1 in the code to compensate.
3. Next: PSN
4. Atomic send: VCU118 requestor (going out VCU118). Then Mellanox NIC will receive the packet. Mellanox NIC. If the test is less than 320.

Week 1 (5/2)

1. atomic send operation: from work request to QSFP. coding the QP request.
2. next task: pull clean firmware to test in blkrock HP. testing of atomic. FPGA is receiving, new code is atomic going out of QSFP of FPGA.
3. on the transmit side, since the information is from the host. from WQE, we don’t have the information.

April 2023

Week 4 (04/25)

1. RISC-V interrupt table of registers with names and address
2. RISC-V interrupts specification
3. Added firmware qp_wr qp work request task, to process work requests from the queue.
    - MLX5_OPCODE_UMR:
    - MLX5_OPCODE_NOP:
    - MLX5_OPCODE_RDMA_WRITE:
    - MLX5_OPCODE_RDMA_WRITE_IMM:
    - MLX5_OPCODE_SEND_INVAL:
    - MLX5_OPCODE_SEND:
    - MLX5_OPCODE_SEND_IMM:
    - MLX5_OPCODE_RDMA_READ - ongoing coding
    - MLX5_OPCODE_ATOMIC_CS: - ongoing coding
    - MLX5_OPCODE_ATOMIC_FA: - ongoing coding
4. checking structure of swap_addr, compr_addr, not detailed in document (document only say swap and compare). Need to reverse engineer the structure and the value. Previously the atomic structure was totally different from write. Every WQE they are different. The wrt.wqe segment is different from atomic. Remote_address_data_seqment is needed for atomic data structure. Need swap address, compare address. Unfortunately, the documentation does not have a definition of the structure and lacks information on the structure. Using user space library Mellanox which bypass kernel driver.
5. **PA_MR** is only supported by OFED 4.9x. Since we are using 5.5 we need to downgrade the older version. OFED 4.9 has compatibility issues.
6. **DC (dynamic connection)** will use processes from atomic operation. Need additional packet needed because it's dynamic.
7. Ethernet side/TCP: FPGA to Mlnx parsing Ethernet packet.
    a. FPGA will RDMA get the EThernet packet from remote
    b. FPGA check if it's RoCE/Infiniband packet
    c. if RoCE - process in Infiniband, else if Ethernet will process as SQ/RQ
    d. RoCE - Que Pair, RQ, Share RQ
    e. RX: via QSPF packet comes to FPGA
    f. Determine opcode
    g. IPV4 parser - we coded the firmware. TX: our firmware creates checksum because Mlnx verifies checksum.
    h. Read packet - check command read, write, send
    i. Memory region translation -using the key
    j. DMA data to local
    k. Transmission - going out from FPGA
        i. from WQE
        ii. Create Virtual address

iii. Create Remote Key
iv. Swap (or Add) data
v. Responder: Mlnx

Week 3 (04/18)

1. Edit Chapter 18, firmware architecture
2. Edit Chapter 5, BAR0 corrected address
3. update non-existing link to QP chapter
4. update status register and configuration Bram port from 7 to 11 of chapter 11 section 2
5. fixed typo errors and proper numberings
6. PA-MR:
    a. Use RDMA experimental verbs library, which is only supported in mlnx-ofed 4.9.x while the system of blockrock and hp are using 5.5
    b. DC connection && Atomic tx-pkt
        i. added firmware to create a received DC connection packet.
        ii. added and structure firmware to create tx-pkt with consideration on other wqe command "atomic"
7. testing to be conducted this coming thursday
8. re-structure the firmware to avoid too many ifs

Week 2 (04/11)

merged code between latest and process management branch

1. mlx5_fw.c
    a. modify function
        i. run_mlx5_fw
        ii. qp_reg_mr
        iii. transfer recv_data from mlx5_fw.c to net.c
        iv. qp_recv_pkt
        v. create_cqe
        vi. uart_bf_dbs
        vii. qp_process
    b. add function
        i. run_mlx5_fw_int_done
        ii. h2c_dma_done and c2h_dma_done
        iii. h2c_get_wait and c2h_get_wait
        iv. h2c_activate and c2h_activate
2. mlx5_fw.h

    a. add function prototypes
3. created pa_mr?, both ubuntu and centos are able to compile
    a. register the physical address to virtual?
    b. physical address becomes contiguous?
    c. the device receiving the transfer will save one function to manage the translation
    d. physical address is the memory from the host
4. looking how to test, one of the .. extended mr, its not
5. need to re compile the library so that the experimental …. saw in the nvidia
    a. if we want about the physical region, the … service
    b. saw some test, not part of mellanox from berkeley, tried to copy and run the code and see the behavior?
6. looking right now how to test ..
7. need to modify the ibm core?

Week 1 (04/04)

merged code between latest and process management branch

1. rdma_opcode.h
    a. add __packed to
        i. r_bth struct r_deth r_immt
        ii. r_atmack r_ieth r_deth
        iii. r_reth r_aeth ib_pkt_struct
        iv. icrc_struct
2. rdma_opcode.c
    a. add include
        i. stdio.h stdint.h string.h
        ii. machine/_type.h
3. qp.h
    a. add
        i. macro struct rgid_rip_struct
        ii. qp_pkt_struct variables to qp_ctrl
        iii. prototype qp_mb_modify_ctx modify qp_wr_fetch_wqe
        iv. modify qp_recv_prcs
4. qp.c
    a. add
        i. include net.h
    b. function
        i. qp_get_rmac qp_get_rgid_rip get_recv_ack
        ii. init_tx_pkt init_bth_hdr cacl_icrc
        iii. qp_recv_atomic qp_recv_pkt qp_recv_prcs
        iv. qp_wr_bth_init qp_req dbg_crc_check
    c. modify

        i. qp_recv_rdma_wr_function qp_wr_fetch_wqe
        ii. qp_wr_req qp_reg_mr qp_mb_create_qp_ctx
        iii. qp_mb_modify_ctx qp_ctrl_init
5. mlx5_fw.h
    a. add enum
6. mlx5_fw.c
    a. add
        i. mlx5_cap_.h net.h srq.h
        ii. sqrq.h mkey.h fetch_mbox_data
    b. modify
        i. hca_cap_gen array
7. Shared RQ (SRQ) - Done, Tested between VCU118 and mlx5 CX5
8. PA-MR needs to recompile the kernel MLNX-OFED enable flag --with-pa-mr
    a. Tried to follow instructions from nvidia docs to recompile the MLNX-OFED but in RPM.
    b. Looking for reference on how to recompile the MLNX-OFED in debian linux

March 2023

Week 4 (03/28)

Question to Kevin:
Manage pages
Huge table that have a memory region
Hi Kevin, just to confirm if the PA_MR is related to the manage pages command from Mellanox? Can we ask more details about it because we didn't find the PA_MR in the Mellanox PRM?
PA_MR is a Physical Address Memory Region.

1. changed xmit_data implementation using 3 parameters
2. add qp_recv_atomic
3. copy the project file in blockrock to kdiff the differences locally
4. Coded firmware CQ.c and CQ.h for modularizing firmware structure.
5. added firmware SRQ.c and SRQ.h for srq module capability.

   - Kernel did not print the cqe content because the test is happening in user space.

   - Trace perftest and rdma-core source code.

   - Debugging completion for using ib_send_bw,

   - Enable debug mode of rdma-core for MLX5_DEBUG option

   - Added debug print on perftest and rdma-core user libraries to print the cqe content

   - Issue Debug print did not print when running the ib_send_bw.

Week 3 (03/21)

1. Added newly supported WQE Opcode
   a. ATOMIC Acknowledge
   b. CmpSwp
   c. FetchAdd
2. Added BAR0 FIFO
   a. BRAM9
   b. BRAMA
   c. BRAMB
3. Update Current Memory Map Figure 8.1
4. Update Chapter 10.3
5. Add Figure 10.3.1 and BAR0 FIFO Diagram and Definition
6. SRQ:
   a. driver will send mail box

b. srq_idx: in real the SendCreate
c. in the create sqrq, get the context, get the size and return the value of srq.
d. how to use the context - connect the srq.
e. the driver stash the number of pages. we need to put a function response to the switch case. From initialization, there is a handler. Need to add routine to include fetching of physical address to keep inside HCA. After fetching that, we can return.

Week 2 (03/14)

1. compile code using shared setup (741866422)
2. compile error.
    a. trying to port only the files that have been modified for task management to improve performance. mlx5_fw.c, mlx5_fw.h, qp.c, and qp.h
    b. It seems there are outdated functions that need to be adjusted like xmit_data, before the parameter was only 2, right now, the parameters are 3. and other functions.
3. to continue compiling and fix the error

Week 1 (03/07)

1. to recompile in tested remote vitis
2. for remote testing/debugging
3. to check performance gain
4. done: generation of invariant crc (icrc) for infiniband header. magic number (0xDEBB20E3)
5. The VCU118 successfully acknowledge the atomic packet from mlx5-CX5
6. added firmware to structure the qp context and network header
7. added firmware to get the remote mac and remote ip from qp context structure.
8. added firmware structure to vport_nic_context.

February 2023

Week 4 (02/28)

1. vitis compiling
2. fixed h2c_activate/c2h_activate
3. fixed h2c_dma_done/c2h_dma_done
4. fixed mainloop
5. for actual testing
6. issue : generation of crc
7. achieved : response is now working
8. Atomic fetch Add:

   VCU118 Fpga: Responder "ib_atomic_bw -d mlx5_0 -o 1 -p 19876"

   Mlx5-CX5 NIC: Requester "ib_atomic_bw -d mlx5_0 -o 1 -p 19876 192.168.5.100"

   VCU118 is able to acknowledge the atomic request from mlx5-CX5, but the Mlx5-CX5 encountered an error with an error message.

   " Completion with error at client

   Failed status 12: wr_id 0 syndrom 0x81

   scnt=128, ccnt=0

   Error occurred in run_iter function"

   Possible issue ICRC since the VCU118 create a packet with 0 ICRC and wrong ICRC

9. Added firmware to compute checksum for IPv4

   Added firmware to compute ICRC, but return a wrong result

10. Ongoing: To find the sequence and right computation of ICRC.

Week 3 (02/21)

1. continue vitis compiling for circleQ usage (task management)
   a. mlx5_fw.c
      i. fixed recv_data
      ii. fixed xmit_data
      iii. fixed flush_data
      iv. fixed fetch_data
2. Able to key exchange using TCP protocol between mlx5-CX5 and VCU118 FPGA
3. Fixed Invalid argument when the ibv_modify_qp to init2rtr is issued.

4. added firmware grh header structure in net.h
5. Ongoing : update firmware mailbox cmd_modify_qp_init2rtr to store GID and mac address. modify_qp need to …
6. all set bits corresponding to modify_qp need to be set from 111 to hp data transfer, test sample (blockrock to …)
7. in order to create key exchange, need to create a packet …create a packet based on wqe right now, the qpn can be sent to the other side can also use the atomic standard command the header that i've created is a double header
8. the response request don't need to … in vcu i know the source but the i didn't the target using pingpong i am going to make atomic rdma sent - pingpong sync
9. need to talk to mellanox in a standard way, need to use the standard structure GAD - Global address
10. atomic exchange sequence - create qp, exchange the qp, exchange the key, modify qp context, after that, it will depend on what command (rdma read, write, etc)

Week 2 (02/14)

1. vitis compiling for the additional circleQ usage (task management)
    a. mlx5_fw.c
        i. remove sll_waittaskinit function (not needed)
        ii. remove unused local variable declaration
        iii. correct type casting from unsigned int to pointer and vice versa
        iv. correct usage of header files "task.h
2. make the tcp work and atomic task
    a. make the tcp work (connection of 2 network is not successfully created)
        i. based on research, the one they saw
    b. on the atomic - getting an error invalid value. before it was not supported, now its invalid value. The problem that I had before is not supported. Upon checking the mellanox, in capabilities, we tell that we supported the atomic, we already tell them that we already atomic, atomic swap.
    c. in initialization right now, it's already supported
    d. there is second query hca
    e. checking right now, is verifying what is the value I sent
    f. what 2 and 22 represent, ibv modify_qp , this is the error code 22, invalid argument
    g. in atomic operation, first you are going to create a qp, once you create that one you are going to put it in initial state, ibv_qp_init, initialize the qp that is already created
    h. to modify again the qp that initialize, init_to_rtr the function has the modified qp if its atomic or the access
    i. right now, this is where we get the error, on the second part of init initialization

Week 1 (02/07)

1. Fixed error message "Couldn't Create MAD cq".
2. Modified firmware mk_translate to include the stride idx
3. Look where the standard key exchange happen: Key exchange happen using the TCP protocol same as rsync
4. local address: LID 0x0000, QPN 0x000255, PSN 0x3cdd5e, GID fe80::ee0d:9aff:fe20:e490
   remote address: LID 0x0000, QPN 0x000256, PSN 0x4113fc, GID fe80::ee0d:9aff:fe20:e491
5. The fpga NIC can ping the mlx5-CX5 but with an error in data
6. bill@HP:~$ ping 192.168.5.111 PING 192.168.5.111 (192.168.5.111) 56(84) bytes of data. 64 bytes from 192.168.5.111: icmp_seq=1 ttl=64 time=1.98 ms
7. wrong data byte #54 should be 0x36 but was 0x0 #16 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f #48 30 31 32 33 34 35 0 0 64 bytes from 192.168.5.111: icmp_seq=2 ttl=64 time=1.95 ms
8. wrong data byte #54 should be 0x36 but was 0x0 #16 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f #48 30 31 32 33 34 35 0 0 64 bytes from 192.168.5.111: icmp_seq=3 ttl=64 time=1.88 ms
9. wrong data byte #54 should be 0x36 but was 0x0 #16 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f #48 30 31 32 33 34 35 0 0 64 bytes from 192.168.5.111: icmp_seq=4 ttl=64 time=1.91 ms
10. wrong data byte #54 should be 0x36 but was 0x0 #16 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f #48 30 31 32 33 34 35 0 0
11. changed command manager implementation with circular doubly linked list (17.2)
12. Added API list
    a. task_init
    b. get_task_entry
    c. return_task_entry
    d. task_put_runQ
    e. exec_task_entry
    f. task_get_head_runQ
    g. task_put_to_sleepQ
    h. task_get_from_sleepQ
    i. task_exec_runQ
13. add firmware implementation using process management and circular doubly linked list
14. started porting firmware to vitis ide for compiling - ongoing

January 2023

Week 4 (01/31)

1. fixed h2c_activate (read dma) functions by removing exec_task_entry and adding Singly Linked List algorithm when there is no more room for wait_task. this fixed is also applied c2h_activate (write_dma)
2. add algorithm by using Singly Linked List during h2c_dma_done when there is data to be processed in wait_task delayed. this addition is also applied in c2h_dma_done
3. The packet data received from the network is now visible to the host using wireshark.
4. Changes: use available pages in rqn in circular approach, one packet one page and update the memory translation function.
5. There some received data that are not yet visible in the wireshare like the ARP response. - ongoing investigation.
6. looking on rsync, what command is expected to receive from the host to the device.
7. studying what is the right approach relating to translating table (stripe, size, packet size)
8. Investigate error regarding create_cq of ib mad.

Week 3 (01/24)

1. task management (task.c) application
    a. run_mlx5_fw
        i. query_hca_cap
        ii. set_hca_cap
        iii. query_special_ctx
        iv. query_nic_vport_ctx
        v. query_adapter
        vi. create_eq
        vii. create_cq
        viii. create_mkey
        ix. create_mod_qp
    b. wqe_process
        i. wqe_opcode
            1. reg_mr
                a. qp_reg_mr
                    i. qp_reg_mr_intr_done
                    ii. reg_mr_cont
                    iii. reg_mr_cont_intr_done
                    iv. create_cqe
                        1. create_cqe_intr_done
                        2. creaate_cqe_intr_done_wr

3. event_hndl_create_cqe
    a. event_hndl_create_cqe_int_done
2. qp_process
    a. qp_wr_fetch_wqe
        i. qp_wr_fetch_wqe_intr_done

c. recv_process
    i. qp_recv_pkt
        1. qp_recv_rdma_wr
            a. qp_wr_fetch_wqe
d. create_cqe
e. qp_process

2. The device can respond using the created CQE for recv packet from net dma.
3. The recv wqe is compose of wqe ctrl segment (16B) and Recv data segment (16B)
4. To isolate the use of the buffer from wqe receive data segment, limit the page to one, just to confirm if the data from the device is visible to the host via wireshark.
5. Issue: The packet data recv from network and transfer to host is (zero) not seen on the host side using wireshark
6. debugging the driver how wqe rsq are being used by the host
7. in mellanox, the approach is the size and mkey and the addr offset, there is relationship of the mkey, right now, this is what i observed
8. richard) -
9. marvin) - dma 1 message then checked if its visible from the host
10. richard) - did the host receive any interrupt?
11. marvin) - it received but the data is not there
12. richard) - before you receive any info do you have a resource to receive it?
13. marvin) - yeah it is setup but the actual information cannot be seen
14. compiling using blockrock is almost the same
15. some of the applications after initializing …
16. the communication is between host and the card
17. The only issue is the receive part. from host to device

Week 2 (01/17)

1. task management (task.c) application
    a. run_mlx5_fw
        i. query_hca_cap
        ii. set_hca_cap
        iii. query_special_ctx
        iv. query_nic_vport_ctx
        v. query_adapter
        vi. create_eq
        vii. create_cq
        viii. create_mkey
        ix. create_mod_qp

    b. wqe_process
        i. wqe_opcode
            1. reg_mr
                a. qp_reg_mr
            2. qp_process
                a. qp_wr_fetch_wqe
                    i. qp_wr_fetch_wqe_intr_done
2. added firmware for receiving incoming packet from rx dma
3. Place received packet to buffer from rxq.
4. the buffer offset from wqe rq is
5. The allocated buffer from wqe rq, is big, that the firmware is not able to handle in the translation table.
    - move mkey to a different memory section (0xc0080000) in ld. Still not fit.
    - changed log max hca capability to make it smaller. The driver did not follow the set max size
    - modify mlx5 driver to limit the maximum allocated buffer. Driver crashes after changes.
6. temporary fix: skip the buffer that does not fit on the translation table.