Behavioral Details of WUGS Switch Chips

J. Andrew Fingerhut

Washington University
Saint Louis, Missouri

Other Information Sources

- All literature citations in this document are to the bibliography in the System Architecture Document, and use the same abbreviations as those used there. Citations to the System Architecture Document itself all appear as "[SAD]".
  - Here are bibliography entries cited here that do not appear in [SAD].

[Turn98] Turner, Jonathan S., "Gigabit Kits Course Switch Architecture," Slide presentation given Summer 1998 to Gigabit Kit participants. http://www.arl.wustl.edu/~jst/gigatech/kits.html

[MR89] Melen, Riccardo and Jonathan S. Turner, "Nonblocking Multirate Networks," SIAM J. Computing. March 1989. http://www.arl.wustl.edu/~jst/pubs/siamjc89.ps (web version does not contain figures)

[MR93] Melen, Riccardo and Jonathan S. Turner, "Nonblocking Multirate Distribution Networks," IEEE Trans. Communications. Vol. 41, No. 2, pp. 362-369. February 1993. http://www.arl.wustl.edu/~jst/pubs/ieee293a.ps

[TY98] Turner, Jonathan S. and Naoki Yamanaka, "Architectural Choices in Large Scale ATM Switches," IEICE Transactions, 1998. http://www.arl.wustl.edu/~jst/pubs/ieice98.ps

- For more information on:
  - Acronyms: [SAD, Sections 6.1, 6.3, 6.4, and 7.1] - sorry, no glossary (yet)
  - Control cell operation code (OPC) values: [SAD, Figure 22]
  - Control cell return values (RVAL): [SAD, Figure 25]
  - Format and meaning of bits in INFO field for reading and writing VXT entries: [Figure 25 in this document, and SAD Figure 27 and Section 7.1]
  - Format and meaning of bits in INFO field for reading and writing maintenance register fields: [SAD, Section 7.2]
  - Known problems and possibly surprising features of the WUGS-20: [SAD, Section 11]
  - The interface provided to link adaptor cards by the WUGS-20: [RF-94a]
- Some notation:
  - "/=" is VHDL for "not equal to".
  - A[i] indicates bit i of the value A. A[i:j] indicates bits i down to j of A. (This is not VHDL's notation, but Verilog's. It is more compact.)

Road Map

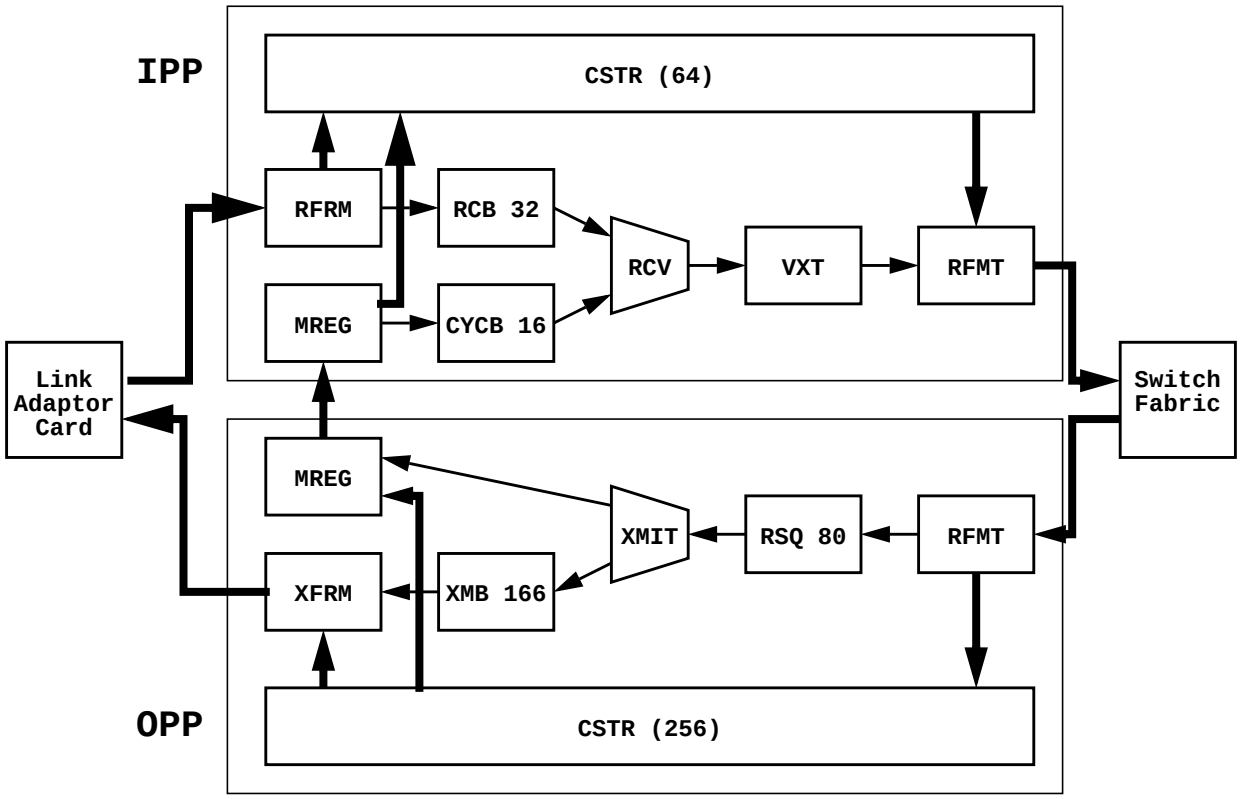


Figure 1: IPP and OPP Block Diagrams

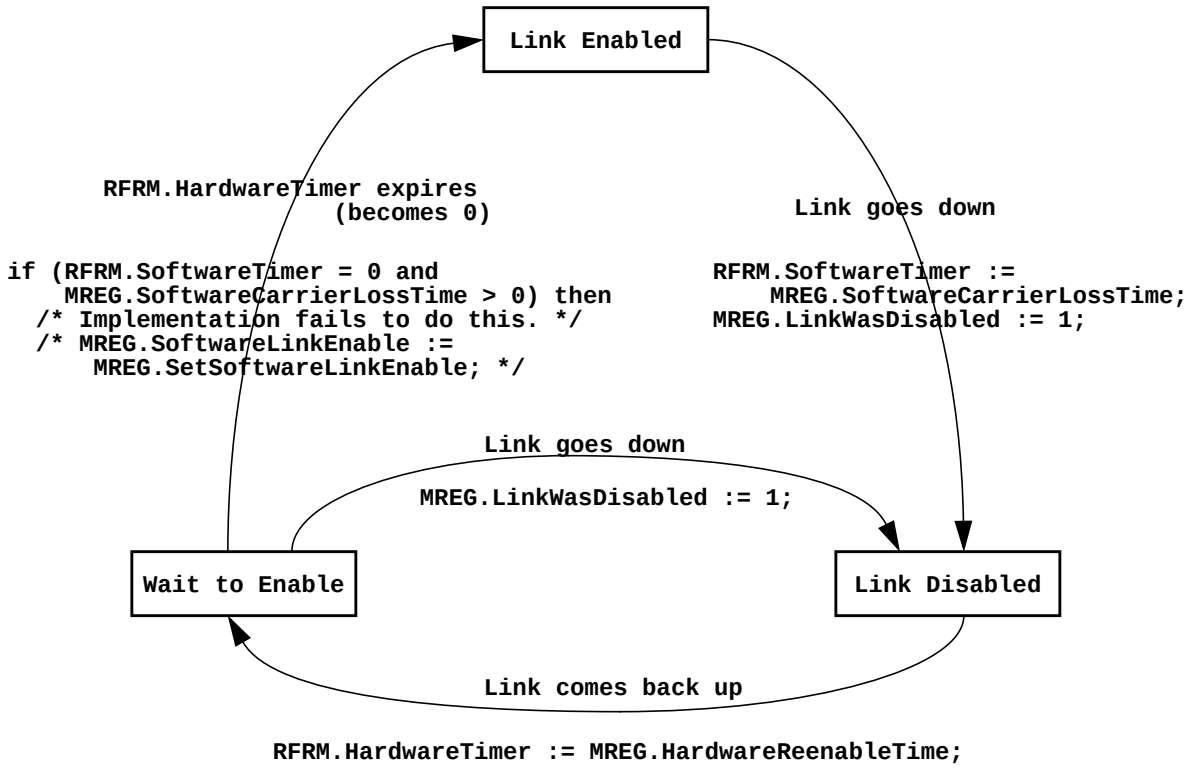
- Numbers indicate capacity in cells.
- Straight through path of IPP
  - RFRM → RCB → RCV (skipped - too simple) → VXT → RFMT
- Straight through path of OPP
  - RFMT → RSQ → XMIT → XMB → XFRM
- Recycling path
  - OPP MREG → IPP MREG → CYCB
- Switch fabric

IPP RFRM Link Enable/Disable Function

- Purpose
  - Disable link when link adaptor card indicates loss of signal (usually because cable was unplugged)
  - Re-enable link in controlled fashion (deal with intermittent signal presence during transition)
- State Machine
  - 3 states: Link_Enabled, Link_Disabled, Wait_to_Enable
  - go to Link_Disabled state when link goes down
  - go to Wait_to_Enable state when link comes back up
  - return to Link_Enabled state following timeout
- Intended software timer function
  - Control software might wish to configure a port to automatically disable data traffic if someone unplugs a cable and then plugs in a different one. This could allow control software to monitor the network topology.
  - The hardware can’t detect this situation directly, but it can detect when the link has been disabled for a “long” time, say 5 seconds. If the link has been down for a long time, then optionally disable all data cell forwarding for that input port.
  - This feature does not work in IPP version 1 due to an error in the maintenance register. Fixed in IPP version 2.

IPP RFRM (Receive Framer) Link Enable State Machine

- Maintains IPP's HardwareLinkEnable state bit.
  - Goes down as soon as link adaptor card is down.
  - Waits for configurable time after link adaptor card indicates it is up before allowing cells through, in hopes of avoiding "garbage" that might be present then.




```
constant LINK_UP := 0;

if (IPPpin.WIDTH_LINK = MODE_32BIT (1))
      and (IPPpin.D_SKEW_LINK = 1) then
   linkup := (IPPpin.UP_L_LINK = LINK_UP)
               and (IPPpin.UP_H_LINK = LINK_UP);
else
   linkup := (IPPpin.UP_L_LINK = LINK_UP);
end if;
```

Name	MREG field #	value after chip reset	length (bits)	RFRM access
HardwareReenable-Time	2	2^{20} cell times ≈ 0.24 sec at 70 MHz	32	read only
HardwareLinkEnable	3	current link status (1 for up)	1	Write every cell time, so control cell writes only take effect for that long.
LinkWasDisabled	3 group, 7 indi-vidual	1 (0 while IPP RESET pin is asserted, but changes to 1 soon after RESET deas-serted because Hard-wareLinkEnable is kept 0 for 2^{20} cell times)	1	set to 1
SoftwareCarrierLoss-Time	2	0	32	read only
SoftwareLinkEnable	2	0 (note well! Must be changed to 1 to enable data cells on port.)	1	read only (due to imple-mentation error)
SetSoftwareLinkEnable	2	0	1	read only

Table 2: MREG fields relevant to IPP RFRM Link Enable State Machine

IPP RFRM (Receive Framer)

- Converts 16 or 32 bit interface to internal 32 bit cell format.
  - Variety of commercially available ATM/SONET framing chips with 16 bit interfaces: OC-3, OC-12
  - Need 32 bit wide interface to implement OC-48 at “comfortable” clock rates (< 80 MHz).
- Discard cells with bad HEC.
- Discard some ATM OAM cell types that are not handled
  - Ideally, they would be forwarded to control processor with full cell header and payload, as well as an indication of which port it was received on. We considered it not worth the design effort to construct such a mechanism.
- For security, discard control cells if control enable jumper on main board is off (separate jumper for each port).
- If control enable is on, initiate entire switch reset or clearing of error flags for control cells with those opcodes.
  - If the chips do get in a bad state, we want such control cells to go through as little circuitry as possible before carrying out their actions.

IPP RFRM Details

- There is no flow control signal from the IPP back to the link adaptor card.
  - If you really, really want one, you could change the IPP TEST_EN input pin to logic 1 (currently hardwired to 0 on the main board), set the RCBDiscardThreshold to some value smaller than the RCB size (see RCB description), and sample an IPP test output that is high whenever the RCB occupancy is over the threshold, synchronizing it to the clock on your adaptor card.

```
if (MREG.HardwareLinkEnable = 0) then
   cell disappears with no state updated;
else if (cell's HEC is wrong) then
   discard cell and increment MREG.BadHECCounter
                              (all counters wrap around to 0);
else if (cell is unassigned cell) then
   discard with no state updated;
else
   increment MREG.ReceiveCellCounter;
   if (cell is GBN switch control cell) then
      if (IPPpin.CTRL_EN = 1) then
         if (control cell opcode is reset or clear error) then
            reset, or clear error flags in, entire switch;
         end if;
         propagate cell;
      else
         MREG.BadControlCell := 1;
         discard cell;
      end if;
   else if (cell is "badsig discard" (see Table 3)) then
      MREG.BadATMSignalingCell := 1;
      MREG.BadATMSignalingCellHeader := 32 bit cell header
                 (all except HEC, which is correct at this point);
      discard cell;
   else if (MREG.SoftwareLinkEnable = 1) then
      propagate data cell;
   else
      discard with no state updated;
   end if;
end if;
```

More IPP RFRM Details

Cell Type	VPI	VCI	PT	CLP	Action
Unassigned Cells	0	0	X X X	0	discard quietly
Segment OAM F4 Flow Cells	X	3	0 X 0	X	badsig discard
End-to-end OAM F4 Flow Cells	X	4	0 X 0	X	badsig discard
Segment OAM F5 Flow Cells	X	≠0	1 0 0	X	badsig discard
GBN Switch Control Cells	0	32 decimal	X X X	X	propagate if CTRL_EN option pin enabled
Everything not covered above treated as data cell					see details

Table 3: ATM cell header fields for different cell types

Name	MREG field #	value after chip reset	length (bits)	RFRM access
BadHECCounter	5	0 cells	32	increment
BadControlCell	3 group, 10 individual	0	1	set to 1
BadATMSignalingCell	3 group, 11 individual	0	1	set to 1
BadATMSignalingCell-Header	3	undefined	32	write only
SoftwareLinkEnable	2	0 (note well! Must be changed to 1 to enable data cells on port.)	1	read only
ReceiveCellCounter	4	0 cells	32	increment

Table 4: MREG fields relevant to IPP RFRM

IPP RCB (Receive Buffer) Function

- Congestion in switch fabric can back up into IPP, causing RCB to fill.
  - RCB can also fill if total of recycling traffic and traffic from link is larger than the bandwidth out of IPP, because recycling traffic has strict priority.
- If RCB fills above congestion threshold, discard arriving low priority (CLP=1) cells and signal congestion to VXT so it can also attempt to clear congestion.

IPP RCB Details

- FIFO with 32 cell capacity.
- Holds only pointer to cell in cell store, plus VPI, VCI, CLP, and control/data cell indication.
- Discard arriving cell if FIFO is full, or if cell is low priority (CLP=1) and occupancy is strictly larger than configurable threshold MREG.RCBDiscardThreshold.
  - Also send congestion indication signal forward to VXT if over threshold.
  - Increment MREG.RCBCLP0OverflowCounter on each CLP=0 cell discard.
  - Increment MREG.RCBCLP1OverflowCounter on each CLP=1 cell discard.
- Any cell placed in FIFO will be forwarded eventually.
- On output side, send out cell if grant received from CYCB, which is given when CYCB is completely empty.

Name	MREG field #	value after chip reset	length (bits)	RCB access
RCBDiscardThreshold	2	32 cells	8	read only
RCBCLP0OverflowCounter	5	0 cells	32	increment
RCBCLP1OverflowCounter	5	0 cells	32	increment

Table 5: MREG fields relevant to IPP RCB

IPP VXT (Virtual Circuit Translation Table) Function

- Look up output port(s), outgoing VPI/VCI, and other connection options for all data cells, e.g., continuous/discrete stream (CS), bypass resequencer (BR).
- Handle virtual paths as well as virtual circuits.
- Maintain cell counter for each connection.
- For setting up, tearing down, and modifying connections, perform read and write operations on VXT entries in response to control cells.
- For security, discard data cells for connections that control software has not set up.
- If RCB signals congestion, then discard cells in low priority connections (CS=0).
  - This is done for a configurable duration, to have significant impact in reducing congestion, and to concentrate cell loss into fewer packets.
  - We would prefer to discard cells with CS=0 in RCB, too, but CS is not known until cell reaches the VXT.

IPP VXT Details

- Adjustable number of virtual path entries, from 1 to 256.
  - The rest of the 1024 entries are for virtual circuits.
- Shared virtual circuit table means that virtual circuits with different VPI must also have different VCI.
  - This limitation has been removed from IPP version 2 design.
- For more information on:
  - Control cell operation code (OPC) values: [SAD, Figure 22]
  - Format and meaning of bits in INFO field for reading and writing VXT entries: [Figure 25 in this document, and SAD Figure 27 and Section 7.1]

Name	MREG field #	value after chip reset	length (bits)	VXT access
VPCount	2	255	8	read only
RCBDiscardHoldDuration	2	0 cell times	16	read only
VXIOutOfRange	3 group, 9 individual	0	1	set to 1
VXIOutOfRangeHeader	3	undefined	24	write only
VXTCS0DiscardCounter	4	0 cells	32	increment

Table 6: MREG fields relevant to IPP VXT

IPP VXT Details: Data Cell Processing

```
-- For every data cell received...
maxVPI := MREG.VPCount;
maxVCI := 1022 - MREG.VPCount;
outofrange := false;
if ((8 lsb's of incell.VPI) > maxVPI) then
   outofrange := true;
else
   entry := VXT(incell.VPI);
   vpt := entry.VPT;
   if (vpt = 1) then
      -- virtual circuit
      if (incell.VCI <= maxVCI) then
         entry := VXT(1023 - incell.VCI);
      else
         outofrange := true;
      end if;
   end if;
end if;
if (outofrange) then
   discard cell;
   MREG.VXIOutOfRange := 1;
   MREG.VXIOutOfRangeHeader := 8 lsb's of incell.VPI, all VCI;
else if (entry.BI=0 or (cell from link and entry.RCO=1)) then
   discard cell with no state updated;
else if (entry.CS = 0 and VXT.HoldTimer /= 0) then
   -- See below for details of the VXT.HoldTimer
   discard cell;
   increment MREG.VXTCS0DiscardCounter;
else
   increment entry.CC;   -- "CC" is cell count
   propagate cell with fields from entry, except:
   outcell.CLP := incell.CLP or entry.SC;  --force CLP=1 if SC=1

   outcell.VPI1 := entry.VPI1;  outcell.VPI2 := entry.VPI2;
   if (vpt = 1) then
      -- Translate VPI and VCI for virtual circuits
      outcell.VCI1 := entry.VCI1;  outcell.VCI2 := entry.VCI2;
   else
      -- Only translate VPI for virtual paths.
      outcell.VCI1 := incell.VCI;  outcell.VCI2 := incell.VCI;
   end if;
end if;
```

IPP VXT Details: Control Cell Processing

```
-- For every control cell received
maxVPI := MREG.VPCount;
maxVCI := 1022 - MREG.VPCount;
if (control cell target is not this VXT) then
   propagate cell unmodified;
else
   if (control cell accesses virtual path entry) then
      tempvpi := 8 msb's of incell.FIELD;
      outofrange := (tempvpi > maxVPI);
      targetaddr := tempvpi;
   else
      tempvci := 16 lsb's of incell.FIELD;
      outofrange := (tempvci > maxVCI);
      targetaddr := 1023 - tempvci;
   end if;
   if (outofrange) then
      outcell.RVAL := BAD_FIELD;
   else
      outcell.RVAL := SUCCESS;
      --
      -- Note that the read or "write & verification read" will be
      -- performed either on the "main part" of the VXT entry,
      -- or the cell count, but not both.  The choice depends
      -- on the opcode of the control cell.
      --
      if (write control cell) then
         VXT(targetaddr) := incell.INFO;
      end if;
      outcell.INFO := VXT(targetaddr);
   end if;
end if;


-- congestion discard state machine
-- state variable VXT.HoldTimer : 16 bit integer;

-- Every cell time...
if (RCB is congested) then
   VXT.HoldTimer := MREG.RCBDiscardHoldDuration;
else if (VXT.HoldTimer /= 0) then
   decrement VXT.HoldTimer;
end if;
```

IPP RFMT (Reformatter)

- For data cells, combine forwarding info from VXT with body of cell from CSTR.
- For control cells that just accessed the VXT, combine result of VXT operation with body of cell from CSTR.
- Time stamp all cells so they may be placed back in order at the target OPP.
  - Maintain state for one connection that is undergoing “transitional/inflated time stamping”, to maintain cell order after removing a receiver from a multicast connection tree. See [Turn98, p. 23, “Avoiding Misordering During Transitions”].
- Store the id of the “trunk group” at which a data cell first arrives to the switch so that the switch can avoid echoing the cell back to this same trunk group.
  - This echo suppression feature (a.k.a. “upstream discard”) is necessary to support scalable many-to-many multicast connections between multiple WUGS switches. See “Suppressing Echo Cells in Many-to-Many Connections” on page 66.

IPP RFMT Details

- Convert data cells from IO/recycling format (Figure 23) to internal format (Figure 22).
- Convert control cells that arrived on link from CP to switch external format (Figure 21) to internal format (Figure 24).
  - Use the first of the three sets of routing information, and shift the other two sets up one position, so that the next set will be used if there is another pass through the switch for this cell.
- Forwards cells only when given grant from downstream SE.

Name	MREG field #	value after chip reset	length (bits)	RFMT access
TrunkGroupIdentifier (tgi in Jammer)	2	0	16 (only 12 used)	read only
TSOffset	2	128	8	read only
Time	1	0	32	read only

Table 7: MREG fields relevant to IPP RFMT

Transitional Time Stamping State Machine

```
-- State variables:

tton : boolean;      -- Is transitional time stamping on now?
                     -- If not, the following are ignored.
ttvp : boolean;      -- If so, is it on for a virtual path?
ttvxi : 24 bits;     -- VPI of connection (also VCI if VC)
tttime : 12 bits;    -- If on, inflated time value to assign to
                     --    next cell in connection.

-- Every cell time...
outcell.TS := (MREG.Time[10:0]).0;  -- default to current time
if (cell is control cell) then
   if (cell opcode is one
         that initiates transitional time stamping) then
      tton := true;
      tttime := (MREG.Time[10:0]).0 + (MREG.TSOffset).0;
      ttvp := (cell.OPC = WRVPXTTR);
      ttvxi := cell.FIELD;
   end if;
else   -- data cell
   invxi := vxi of cell that it had before VXT translation;
   if (tton) then
      if (ttvp) then
         -- only the VPI must match
         match := (invxi[23:16] = ttvxi[23:16]);
      else
         -- both VPI and VCI must match
         match := (invxi[23:0] = ttvxi[23:0]);
      end if;
      if (match) then
         outcell.TS := tttime;    -- give cell inflated time stamp
         tttime := tttime + 0.1₂;   -- increment by 1/2
      end if;
   end if;
end if;
if (tton) then
   if (MREG.Time[10:0] = integer part of tttime) then
      -- ... real time has caught up with transitional/inflated
      -- time.  Turn off transitional time stamping.
      tton := false;
   end if;
end if;
```

OPP RFMT (Reformatter)

- Every data cell has two sets of the following fields in the internal format, one for each of the two copies of a cell in a binary copy: VPI, VCI, BDI, CYC, UD. The RFMT selects between these based on the RC field.
- Convert data cells from internal (Figure 22) to IO/recycling format (Figure 23).
- Convert link-bound control cells from internal (Figure 24) to Switch to CP external format (Figure 21).
- Discards “echo” cells (see “Suppressing Echo Cells in Many-to-Many Connections” on page 66 for motivation).
- Check parity independently on four planes of the switching fabric.

Name	MREG field #	value after chip reset	length (bits)	RFMT access
TrunkGroupIdentifier (tgi in Jammer)	13	0	12	read only
ReliableMulticast	13	0	1	read only
ParityErrorPlane3	14 / 18	0	1	set to 1
ParityErrorPlane2	14 / 19			
ParityErrorPlane1	14 / 20			
ParityErrorPlane0	14 / 21			

Table 8: MREG fields relevant to OPP RFMT

OPP RFMT Details

```
-- MREG.ReliableMulticast should always be set to 0 when the OPP
-- is used with the IPP version 1.  A value of 1 allows some new
-- types of reliable multicast connections in the IPP version 2
-- to be used.  For more information, see [Turner-96b] for an
-- introduction to the features, [IPP version 2 spec] for all
-- the gory details of its implementation, and [SAD,
-- Section 8.3.1] for the few behavior differences in the OPP
-- RFMT when MREG.ReliableMulticast is 1.

-- For every cell received...
if (incell.RC = “011”) then  -- shouldn’t happen if SEs correct
   discard cell with no state updated;
else if (data cell) then
   if (incell.RC = “001”) then
      outcell.{VXI,BDI,CYC,UD} := incell.{VXI2,BDI2,CYC2,UD2};
   else
      outcell.{VXI,BDI,CYC,UD} := incell.{VXI1,BDI1,CYC1,UD1};
   end if;
   --
   -- If cell is destined for link intf and this OPP is in
   -- the same “trunk group” in which the cell arrived, then
   -- this cell is an “echo cell”.  Discard it if echo
   -- suppression is turned on for the connection (UD=1).
   --
   if (outcell.CYC = 0)     -- cell destined for link intf
      and (outcell.UD = 1)  -- echo suppression is on
      and (incell.STG = MREG.TrunkGroupIdentifier) then
      discard cell with no state updated;
   else
      Put cell into format of Figure 23 and send to CSTR;
      Extract TS, CS, BR, CLP, PT, CYC, BDI, PTR, and send to RSQ;
   end if;
else  -- control cell
   if (incell.RC = “001”) then
      outcell.CYC := incell.CYC2;
   else
      outcell.CYC := incell.CYC1;
   end if;
   Put cell into switch to CP format of Figure 21 if CYC = 0;
end if;
```

OPP RSQ (Resequencer)

- Switching fabric with load balancing of cells within a connection over multiple physical paths can deliver cells to OPP out of order. Resequencer uses time stamps to put them back in order. See [Turn98, p. 22, “Cell Resequencing”].
- Can be “bypassed” on a per-connection basis to reduce latency through switch even further, if desired.
  - Default 60 cell time latency is only 13.7 µs at 70 MHz.
  - For 8 port system, current SE chip is FIFO if consecutive cells in connection arrive at least 8 cell times apart, and cells do not stay in SE longer than 64 cell times. See “SE Operational Details” on page 46 for details.
  - In a multistage switch, can minimize possibility of cell reordering by using specific path routing for point-to-point connections, chosen per connection. This can increase virtual circuit blocking. See [TY98] for summary, [MR93,MR89] for details.
- Age of every arriving cell is calculated based on current MREG.Time and time stamp (TS) field in cell.
  - MREG.Time field of all IPP chips synchronized within clock skew of board. Similarly for OPPs. IPPs sync’d within 2 cell times of OPPs at system reset.
- Up to 80 cells maintained in order via “parallel insertion sort” implementation.

Name	MREG field #	value after chip reset	length (bits)	RSQ access
ResequencerOffset (RsqOffset in Jammer)	13	60 cell times	8	read only
TooLateDiscardCounter	16	0 cells	32	increment
ResequencerOverflowCounter	16	0 cells	32	increment

Table 9: MREG fields relevant to OPP RSQ

OPP RSQ Details

```
-- For every cell received...
if (resequencer is full) then
   discard cell;
   increment MREG.ResequencerOverflowCounter;
else
   if (cell.BR = 1) then
      -- “Bypass” the resequencer. Make age equal to
      -- MREG.ResequencerOffset, regardless of cell’s time stamp.
      age := MREG.ResequencerOffset;
   else
      age := MREG.Time - cell.TS;
   end if;
   --
   -- age is 12 bits, treated as signed two’s complement value in
   -- comparisons below, with “binary point” immediately before
   -- last bit.  MREG.ResequencerOffset is nonnegative integer.
   --
   -- Cell age can legitimately be negative during transitional
   -- time stamping of a connection.  If it is “too negative”, it
   -- must actually be very old and wrapped around to negative
   -- age.
   --
   if (age > MREG.ResequencerOffset) or (age <= -255) then
      discard cell;
      increment MREG.TooLateDiscardCounter;
   else
      Place cell in resequencer, maintaining oldest-cell-first
      order, with the age calculated.  Only 10 lsb’s of age are
      stored for all 80 cells.  This allows ages up to 255.5 cell
      times to be represented.
   end if;
end if;


-- Every cell time...
if (resequencer contains at least one cell
      and oldest cells’s age is >= MREG.ResequencerOffset) then
   send out oldest cell and shift all others ahead one position;
end if;

-- Also every cell time...
increment ages of all cells by 1.0, maxing out at 255.0 or 255.5;
```

OPP XMIT (Transmit Circuit)

- Just a demultiplexor that sends cells destined for the link (CYC=0) to the BDC, and recycling cells (CYC=1) to the MREG.

OPP BDC (Block Discard Control)

- If link adaptor drains OPP at less than the switch's internal data rate (the normal case), flow control can back up into transmit buffer (XMB).
- For bursty data connections using AAL5 framing, it is better to discard entire AAL5 frames, rather than a few cells from each of many AAL5 frames.
- To achieve this goal, the BDC implements a variant of Early Packet Discard with hysteresis (EPDH).
- Early Packet Discard requires at least one bit of state per connection - a propagate/discard state.
  - While this state could be accessed via the cell's outgoing VPI/VCI, naive implementation requires a large sparse table.
  - Hashing not deemed worth implementing.
  - Instead, for every connection passing out the same OPP chip for which you wish to use EPDH, assign the connection a unique 8-bit "block discard index" (BDI).

OPP BDC Details

- BDI=0 indicates a cell in a connection for which EPDH is not used. Such a cell is discarded only when the destination FIFO in the XMB is congested.
- Cells with BDI>0 use that value to access its connection's EPDH state.
  - EPDH state maintained for up to 255 connections per OPP
  - 2 bits of state per connection: "propagate/discard connection" and "next cell is first of its AAL5 frame"
  - EPD works well at preventing partial frame discards if the buffer capacity is at least one frame for each of the active connections. EPDH achieves the same goal with just two frames of buffering, regardless of the number of active connections. See [Turner-96a] for an analysis.

Name	MREG field #	value after chip reset	length (bits)	BDC access
XMBCS0EPDHThreshold	13	67 cells	16	read only
XMBCS0NearEmptyThreshold	13	10 cells	16	read only
XMBCS0OverflowCounter	16	0 cells	32	increment
XMBCS1OverflowCounter	16	0 cells	32	increment
Fields below allow block discard state to be read and written via control cells. See [S.A.D., Section 7.2.2]				
BlockDiscardOperation	22	0	8	read only
BDItoOperate	22	0	8	read only
BlockDiscardStatetoWrite	22	0	8	read only
BDIRead	22	0	8	write only
BlockDiscardStateRead	22	0	8	write only

Table 10: MREG fields relevant to OPP BDC

EPDH Basic Idea and Details

- Basic idea: When the first cell of an AAL5 frame arrives...
  - If the buffer occupancy is over the threshold *and increasing*, the connection is put in DISCARD state.
  - If the buffer occupancy is under (or at) the threshold *and decreasing*, the connection is put in the PROPAGATE state.
  - If the buffer level is below a very small threshold value (e.g., 10 cells), the connection is put in the PROPAGATE state.
  - Otherwise, leave connection in its previous state.

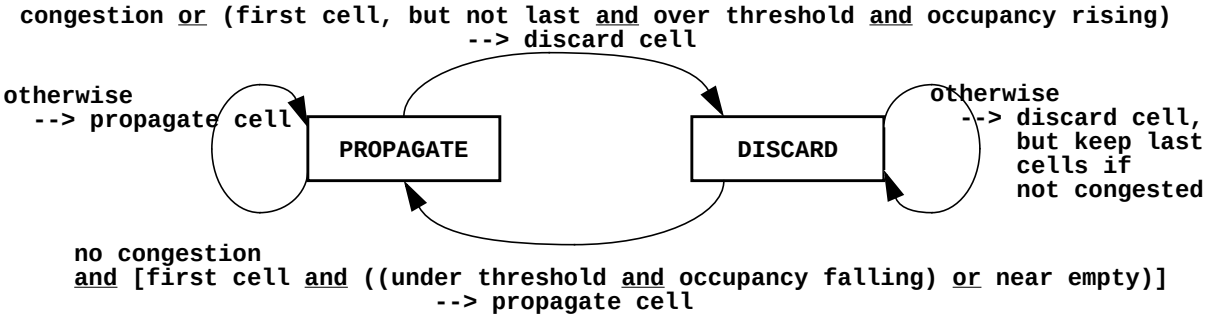


Figure 11: Early Packet Discard with Hysteresis (EPDH) State Machine

- Details
  - In this implementation, EPDH is used for discrete stream connections (CS=0) with BDI /= 0. For continuous stream (CS=1) connections with BDI /= 0, partial packet discard (PPD) is used, where a cell is only discarded if the CS=1 FIFO is full, and then the rest of the AAL5 frame is discarded.
  - Only discard last cells of frames if destination FIFO is congested. Without them, destination reassembly device may concatenate partially discarded frame with next one.
  - Cells with PTI[2]=1 (see [SAD, Figure 16]) bypass this mechanism. They may be ABR resource management cells, for example, which can be mingled within AAL5 frames. They are only discarded when the destination FIFO is congested, and cause no change in the connection's discard state.

More EPDH Details

- Implement “occupancy increasing/decreasing” as follows:

```
-- State variables:
EPDHmin, EPDHmax : integer;

-- every cell time...
threshold := MREG.XMBCS0EPDHThreshold;
occupancy := current XMB CS=0 FIFO occupancy;
if (occupancy > threshold) then
   EPDHmax := max{EPDHmax, occupancy};
   EPDHmin := threshold;
else
   EPDHmax := threshold;
   EPDHmin := min{EPDHmin, occupancy};
end if;
“occupancy increasing” := (occupancy >= EPDHmax);
“occupancy decreasing” := (occupancy <= EPDHmin);
```

```
-- Definition of “congestion”: every cell time...
if (cell.CS = 1) then
   congested := XMB CS=1 FIFO is completely full;
else  -- cell.CS=0
   if (cell.CLP = 1) then  -- low priority
      congested := XMB CS=0 FIFO is full or occupancy >
                          MREG.XMBCS0CongestionThreshold;
   else
      congested := XMB CS=0 FIFO is completely full;
   end if;
end if;
```

OPP XMB (Transmit Buffer)

- Queue cells while waiting for transmission on link.
- Distinguish and give strict priority to high priority connections.

OPP XMB Details

- 166 cell capacity (~8 KB of payloads) split logically into two FIFOs
  - CS=1 cells - “continuous” / low rate variation / CBR, VBR
  - CS=0 cells - “discrete” / bursty / ABR, UBR
  - Logical split point adjustable via MREG.XMBCS0BufferSize at any time with no loss of cells already queued. The current split point may lag behind the specified split point until cells “on the wrong side” are transmitted.
- CS=1 FIFO is given strict priority to leave
- No CS=0, CLP=1 cells allowed to enter CS=0 FIFO when occupancy is over MREG.XMBCS0CongestionThreshold
- Any cell placed in FIFO will not be discarded

Name	MREG field #	value after chip reset	length (bits)	XMB access
XMBCS0BufferSize	13	134 cells	16	read only
XMBCS0CongestionThreshold	13	134 cells	16	read only

Table 12: MREG fields relevant to OPP XMB

OPP XFRM (Transmit Framer)

- Converts internal 32 bit wide format to external 16 or 32 bit wide UTOPIA.
- Only forwards cells when given grant from link adaptor card.
  - See Link Interface Specification [RF-94a] for details on flow control signals.
- Perform explicit forward congestion indication (EFCI) marking
  - Set congestion bit of payload type identifier (PTI[1], see [SAD, Figure 16]) for all user data cells (PTI[2]=0) transmitted while the XMB CS=0 FIFO is congested.
- Increments MREG.TransmitCellCounter once for each cell transmitted.

Name	length (bits)	MREG field #	value after chip reset	XFRM access
TransmitCellCounter	32	15	0 cells	increment

Table 13: MREG fields relevant to OPP XFRM

- Cells with PTI[2]=1 (see [SAD, Figure 16]) bypass this mechanism.

OPP MREG (Maintenance Register)

- Maintains configuration values, error flags, and statistics counters for entire chip.
- Processes control cells that are destined for this particular MREG, and propagates others.
  - Operations provided are read, write, and “respond only if at least one error flag is on”.
  - Most fields can only be accessed in groups, but error flags can be individually addressed so that one may be turned off without inadvertently turning off others in the same group.
- Computes and sends parity for every 32 bit word sent to IPP.
- For more information on:
  - Control cell operation code (OPC) values: [SAD, Figure 22]
  - Control cell return values (RVAL): [SAD, Figure 25]
  - Format and meaning of bits in INFO field for reading and writing maintenance register fields: [SAD, Section 7.2]

Name	MREG field #	value after chip reset	length (bits)	MREG access, in the absence of control cells
ReportErrors	13	1	1	read only
RecyclingCellCounter	15	0 cells	32	increment

Table 14: MREG fields relevant to OPP MREG

```
-- Note: Differences between IPP and OPP MREG are underlined
-- in both descriptions, and marked by arrows in IPP MREG
-- description.

-- For every data cell received...
increment MREG.RecyclingCellCounter;
propagate cell;
```

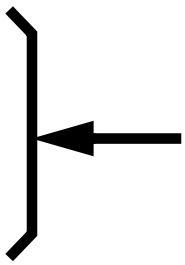
OPP MREG Details: Control Cell Processing

```
-- For every control cell received...
-- Attempt to propagate all control cells except some ERRORS
-- control cells (see below).
increment MREG.RecyclingCellCounter;
temp.AttemptToPropagate := true;
if (cell.COF = 0) then
   cell.LT := MREG.Time;   -- LT = local time
   case cell.OPC is
   when RDMR | WRMR =>
      if (cell.FIELD in {12, ..., 22}) then
         if (cell.OPC = WRMR) then
            Write those fields selected by cell.FIELD that
            can be modified via a control cell with the
            appropriate bits from cell.INFO;
         end if;
         cell.INFO := relevant fields selected by cell.FIELD;
         cell.RVAL := SUCCESS;
      else
         cell.RVAL := BAD_FIELD;
      end if;
   when ERRORS =>
      -- Warning: The pseudocode below does not depend on the
      -- value of cell.FIELD, but the actual implementation
      -- mistakenly does.  Setting it to 0 is safe.
      if (MREG.ReportErrors = 1)
            and (“some error flag is on”) then
         cell.RVAL := SUCCESS; cell.INFO := fields in group 14;
      else
         temp.AttemptToPropagate := false;
      end if;
   when NOP =>    cell.RVAL := SUCCESS;
   when others => cell.RVAL := BAD_OPCODE;
   end case;
end if;
decrement cell.COF, wrapping 0 around to FF hex;
if (temp.AttemptToPropagate) then
   propagate cell;
else
   discard cell;
end if;
```

IPP MREG (Maintenance Register)

- Very similar to OPP MREG, with following differences...
- Recognizes and propagates opcode of control cells destined for VXT.
- Checks parity on arriving 32 bit words every clock period.
- For more information on:
  - Control cell operation code (OPC) values: [SAD, Figure 22]
  - Control cell return values (RVAL): [SAD, Figure 25]
  - Format and meaning of bits in INFO field for reading and writing maintenance register fields: [SAD, Section 7.2]

```
-- For every data cell received...
increment MREG.RecyclingCellCounter;
if (MREG.RecyclingLinkEnable = 0) then
   discard cell;
else if (CYCB's data FIFO is full) then
   discard cell and increment MREG.CYCBDiscardCounter;
else
   propagate cell;
end if;
```



Name	MREG field #	value after chip reset	length (bits)	MREG access, in the absence of control cells
Time	1	0	32	increment, read
ReportErrors	2	1	1	read only
RecyclingLinkEnable	2	1	1	read only
ParityError	3 group, 8 individual	0	1	set to 1
RecyclingCellCounter	4	0	32	increment
CYCBDiscardCounter	5	0	32	increment

Table 15: MREG fields relevant to IPP MREG

```
-- For every control cell received...
-- Attempt to propagate all control cells except some ERRORS
-- control cells (see below).
increment MREG.RecyclingCellCounter;
temp.AttemptToPropagate := true;
if (cell.COF = 0) then
   cell.LT := MREG.Time;   -- LT = local time
   case cell.OPC is
   when RDMR | WRMR =>
      if (cell.FIELD in {1, ..., 11}) then
         if (cell.OPC = WRMR) then
            Write those fields selected by cell.FIELD that
            can be modified via a control cell with the
            appropriate bits from cell.INFO;
         end if;
         cell.INFO := relevant fields selected by cell.FIELD;
         cell.RVAL := SUCCESS;
      else
         cell.RVAL := BAD_FIELD;
      end if;
   when ERRORS =>
      -- Warning: The pseudocode below does not depend on the
      -- value of cell.FIELD, but the actual implementation
      -- mistakenly does.  Setting it to 0 is safe.
      if (MREG.ReportErrors = 1)
            and (“some error flag is on”) then
         cell.RVAL := SUCCESS; cell.INFO := fields in group 3;
      else
         temp.AttemptToPropagate := false;
      end if;
   when “some VXT opcode” =>  -- nothing to do here
   when NOP =>    cell.RVAL := SUCCESS;
   when others => cell.RVAL := BAD_OPCODE;
   end case;
end if;
decrement cell.COF, wrapping 0 around to FF hex;
if (temp.AttemptToPropagate) then
   if (CYCB's control FIFO is full) then
      discard cell;
      increment MREG.CYCBDiscardCounter;
   else
      propagate cell;
   end if;
else
   discard cell;
end if;
```

All differences with OPP MREG are marked by arrows.

IPP CYCB (Recycling Buffer)

- Provides some buffering in the event that switching fabric congestion causes flow control to back up into IPP.
- Contains a control cell FIFO with 2 cell capacity, and data cell FIFO with 16 cell capacity.
  - Control cells must carry 16 byte INFO field for writing VXT entries to set up connections, whereas data cells only need the same CSTR pointer and control information that RCB holds.
- Sends separate full signals for control and data FIFOs back to MREG so that it never sends a control/data cell to full control/ data FIFO, but discards them instead.
- On output side, only sends cell if VXT gives grant.
  - Control cells given strict priority.
  - If both FIFOs are empty, RCB is given a grant to send.

Name	MREG field #	value after chip reset	length (bits)	CYCB access
CYCBDiscardCounter (Jammer leaves out “B”)	5	0	32	MREG increments when CYCB is full

Table 16: MREG fields relevant to IPP CYCB

Other IPP & OPP MREG Fields

Chip	Name	MREG field #	value after chip reset	length (bits)	Comments
IPP	LinkType	1	see comments	8 (only 4 lsb's are defined)	Cannot be modified via software. Contents always reflect values on 4 TYPE_LINK pins from link adaptor card. See [RF-94a, Figure 6] for encoding.
IPP	HardwareReset	3 group, 6 individual	1	1	Software can send control cell to clear this, and then a change indicates that the switch was reset by means other than software control (e.g., power cycled, someone reset the switch with the button on the chassis).
OPP		14 group, 17 individual	1	1	
IPP	ChipType	1	0 (indicating IPP chip)	8	Cannot be modified. Hardwired into chip.
IPP	ChipVersion (Jammer combines this field with previous one)	1	1	8	
OPP	ChipType	12	1 (indicating OPP chip)	8	
OPP	Chip Version (Jammer combines this field with previous one)	12	1	8	

Table 17: Other MREG fields

Switch Fabric

- Route cells from input ports to one, two, or a consecutive range of desired output ports.
  - An arbitrary subset of desired output ports would be nice, but it doesn't scale well to a large number of ports, requiring either routing tables in switch element chips or large bitmaps in cell format. We did not bother implementing a special case for switches with a small number of ports.
- 8 port SE chip designed to implement a nonblocking multistage switch fabric with any number of ports from 8 to 32,768, in powers of two.
  - 32,768 OC-48 ports is 80 Tbps capacity.

Switch Fabric Behavior

- WUGS-20 switch fabric consists of a single 8 port switch element (SE), split into 4 identical chips.
  - Each one receives one fourth of the 32 bit data path, and copies of all 4 of the routing/control pins. The “fourths” of every cell are routed identically and forwarded simultaneously through the switching fabric.
- Grant signals from first stage of SEs to IPPs, and between consecutive stages of SEs, prevents cell loss in switch fabric.
  - Last stage of SEs always receive grants. OPP chips do not send out grants.
  - SEs have 40 cell capacity, and each sends (# unused slots)-8 grants to its 8 upstream neighbors (but no less than 0, no more than 8). If between 0 and 8, they are distributed round robin.
- Switch fabric passes 32 data bits unmodified, without interpretation.
  - Only fields examined: busy/idle (BI), routing control (RC), address (IADR)
    - Due to poor implementation choice, the SE version 1 requires the unused bits to the left of the IADR and Reserved fields to be 0. The IPP fills them in this way.
  - Only fields subject to change: RC, IADR, and Reserved

Switch Fabric Cell Format

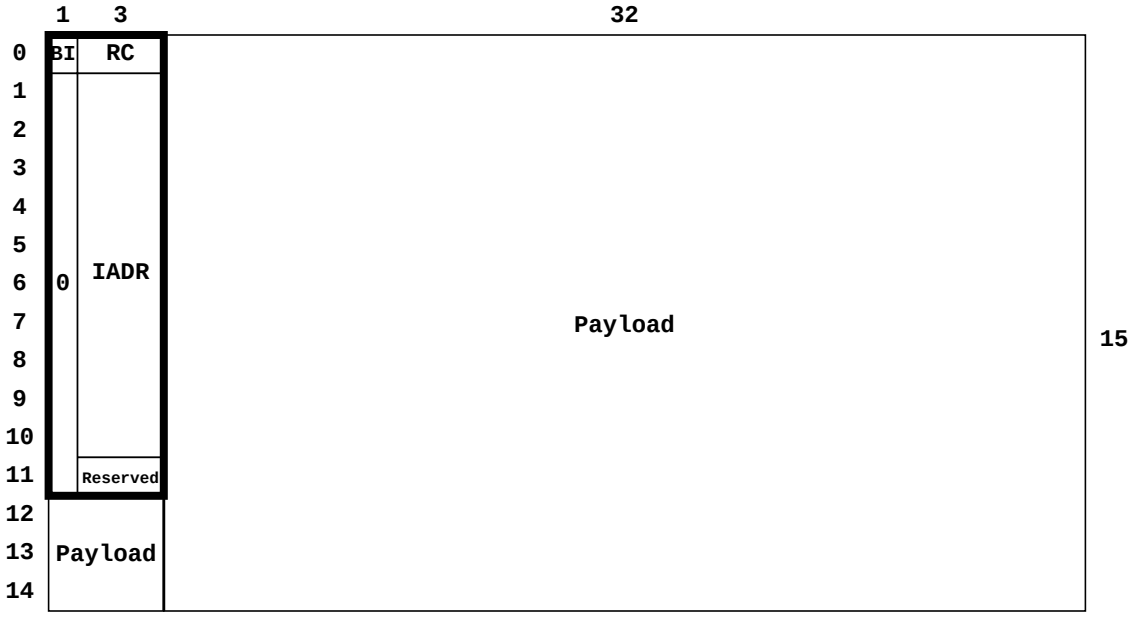


Figure 18: Format of All Cells, According to SE Chips

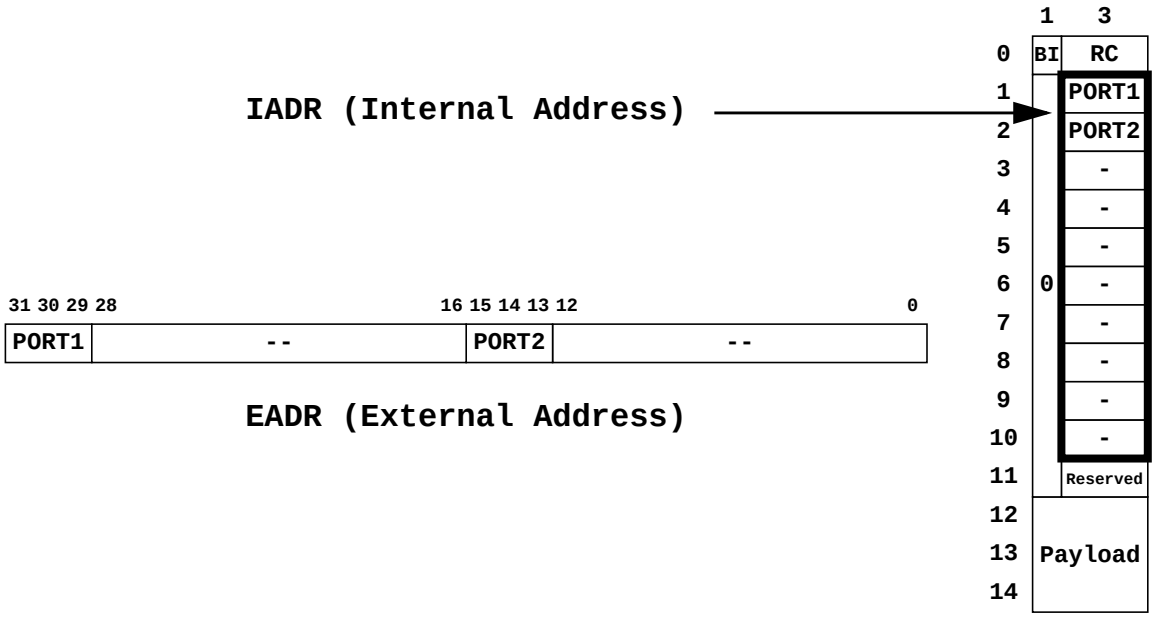


Figure 19: EADR and IADR Format for WUGS-20
See SAD Figure 24 and 42 for general format, in switches with more than 8 ports.

Switch Fabric Routing Options

- Specific path cells ($RC=000_2$): IADR contains a sequence of ten SE output port numbers, used top to bottom, one per stage.
  - Primarily intended for testing internal data paths in multistage fabric, but can also be used for unicast data connections if one wishes to avoid the re-sequencer latency (see OPP RSQ for caveats).
  - IADR is shifted up one row by each stage of SEs passed through, making next output port accessible in same place for all stages.
- Single copy cells ($RC=010_2$ or 001_2) and copy by two cells ($RC=011_2$): IADR contains two row-interleaved switch fabric output port numbers, PORT1 and PORT2, used top to bottom, two rows per routing/copying stage.
  - See Figure 19 for exact format.
  - Cells distributed uniformly without copying in first half of fabric stages, to balance the load across the middle stage SEs. Starting in middle stage, and continuing for the remaining stages, the cell is routed to the desired output port(s).
  - For copy by two cells, cell is copied as late as possible. Two copies are made even if PORT1=PORT2. The copy sent to PORT1 has RC=010, and the copy sent to PORT2 has RC=001, so the OPPs can distinguish the copies.
  - IADR unused and unmodified for first half of fabric. Shifted up two rows by each stage of SEs passed through in last half.
- Copy to range cells (RC=1**): IADR contains PORT1 and PORT2, as for previous case. Fabric makes copies to all ports j with $PORT1 \leq j \leq PORT2$.
  - The implementation requires that $PORT1 \leq PORT2$.
  - As for copy by two, cells are distributed uniformly in first half, and routed and copied in last half, with copying done as late as possible.

SE Operational Details

- Cells are only expected on inputs to which grants were given.
  - If a busy cell arrives on an input to which a grant was denied, it is silently discarded. That should never happen with IPP and SE chip designs.
- Distribution circuits (DSTCs) on input side fill in uniformly distributed output port numbers to arriving cells in distribution stages, except for specific path cells.
  - A cell arriving during cell time *T* at input *I* is assigned output ((*T*+*I*) mod 8).
- All cells, including multicast cells, are stored in exactly one slot of the cell buffer. They are assigned a local age of 0 on arrival, and this age increases by 1 every cell time, up to a maximum of 63, where it “sticks”.
- Every cell time, all cells desiring output *j* compare their ages with each other, with ties broken by the slot number in the cell buffer.
  - Due to time restrictions, only the 3 msb’s are used in the comparison. That is why the SE is FIFO for cells that arrive at least 8 cell times apart, as long as they stay at most 64 cell times.
  - A multicast cell can win contention for an arbitrary subset of its desired outputs, and will never again contend for outputs to which it was previously transmitted.

More SE Operational Details

- If a grant is received from downstream neighbor *j* and at least one cell desires output j, the winning cell is sent to that output.
  - Otherwise a synchronization pattern is sent, for the benefit of the skew compensation circuits at the receiving end.
- Cells are kept in their cell slots until they are successfully transmitted to all outputs they desire. They are transmitted to each desired output exactly once.
- Minimum latency through SE for a cell that wins contention on its first try, and receives a grant from the desired output, is 2 cell times (0.46 μs with 70 MHz clock).
- Since SE chips have no maintenance register, as do IPP and OPP, there is no direct way for software to determine when an SE has detected a parity error on one or more of its inputs.
  - The indirect method we have implemented is that when an SE detects a parity error on input port X, it sets an internal bit to 1 to remember this, and from that point forward it intentionally generates bad parity on any cells sent to output port X.
  - These bits are cleared when the switch is reset, or when all errors are cleared in the switch with an opcode CLR_ERR control cell.
  - The intent is that the parity errors are propagated through the switch fabric until they reach an OPP, where they are recorded in its maintenance register, readable through control cells. Thus, a parity error in the OPP localizes the problem to one of (# stages + 1) data paths in the switch fabric (2 for the WUGS-20).

A Note on Switch and Link Clock Rates

- For a link adaptor card with 32-bit wide data path, the link clock rate must be at most (14/16) times the switch clock rate.
  - The multiplier is (27/16) for a 16-bit mode link adaptor card.
- This restriction is in addition to any given in [RF-94a, Section 5.1].
- Reason: the maximum cell rate of the link must be no faster than the internal switch cell rate, otherwise the IPP receive buffer (RCB) can lose pointers to the cell store, causing a memory leak. Eventually the IPP will lock up until it is reset.
  - If there are guaranteed to be gaps between cells such that the actual cell arrival rate meets this restriction, that is sufficient.
  - The IPP in question should still correctly perform a switch reset if it receives a control cell with the reset opcode, unless the link clock rate is faster than the IPP RFRM's maximum operating rate (this should be about 100 MHz).
  - The link cell rate is $C_{link} = (\varphi_{link} / P_{link})$, where φ_{link} is the link clock rate, and P_{link} is the number of link clock periods in one cell time (P_{link}=14 if link adaptor operates in 32-bit mode, or 27 for 16-bit mode).
  - The internal switch cell rate is $C_{switch} = (\varphi_{switch} / P_{switch})$, where φ_{link} is the switch clock rate, and P_{switch}=16 is the number of switch clock periods in one cell time.
  - IPP RCB requires $C_{link} \leq C_{switch}$.
  - The IPP version 2's RCB has its input bandwidth doubled, so that the restriction for it is $C_{link} \leq 2C_{switch}$.
- In the absence of the RCB restriction, the OPP design has the restriction $C_{link} \leq (\varphi_{switch} / 15)$, at least under heavy loads.

ATM Cell Formats

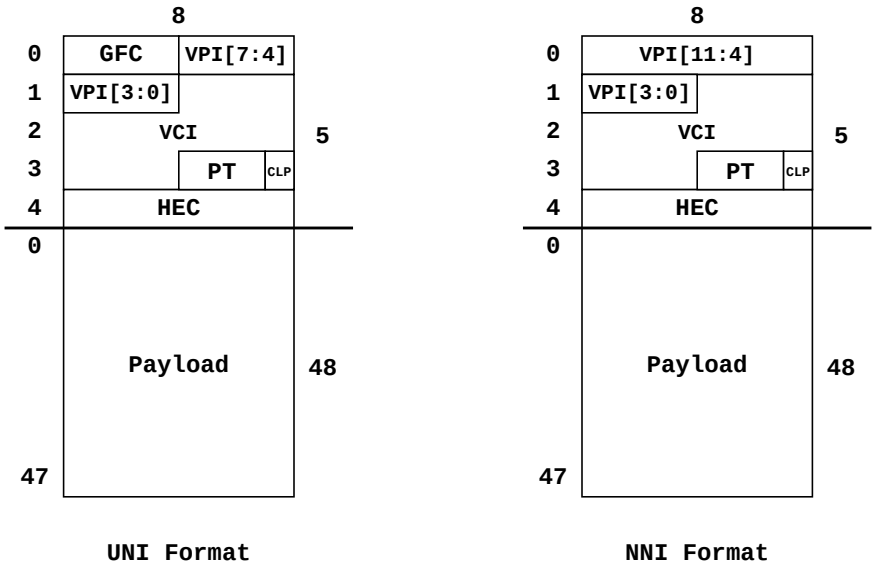
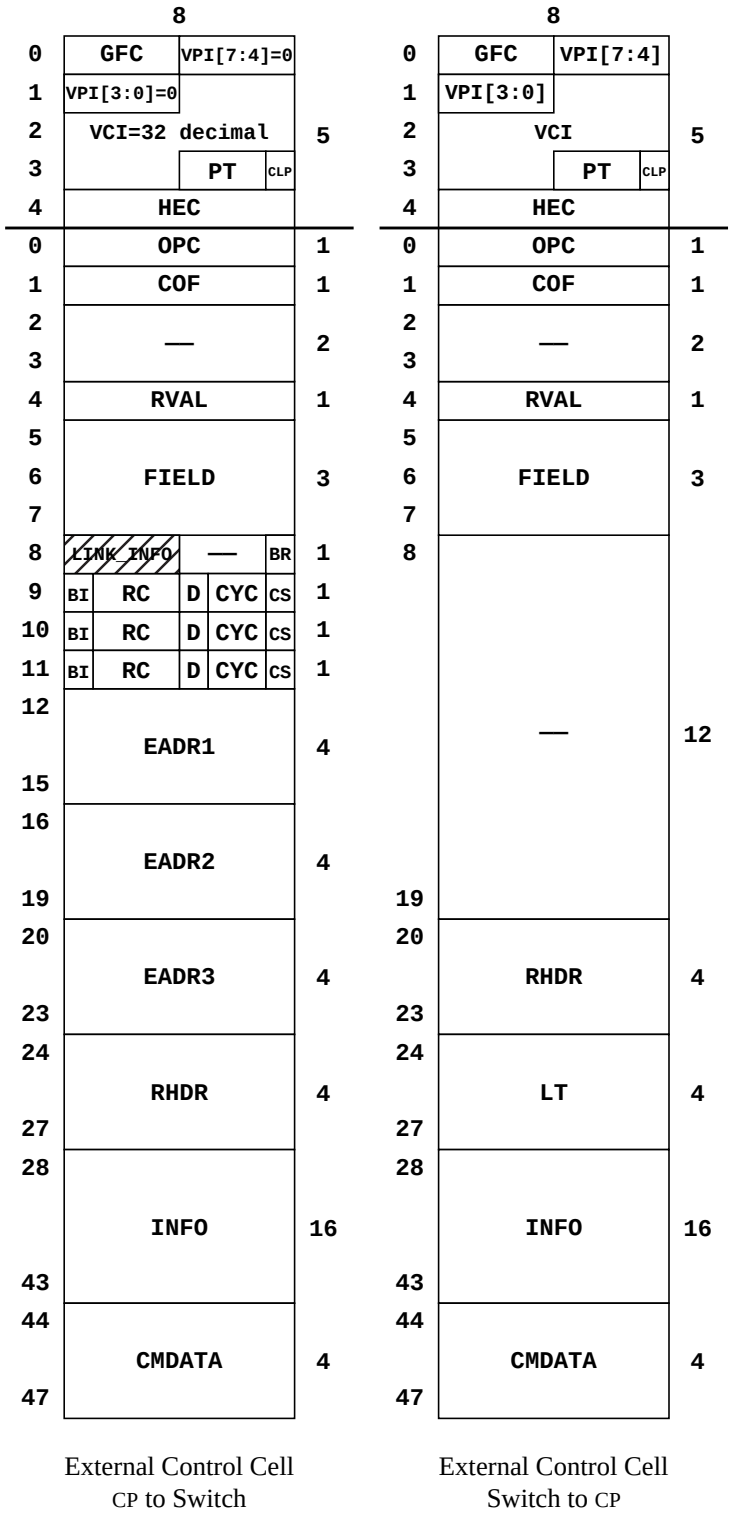


Figure 20: ATM Cell Format

- WUGS ignores GFC/VPI[11:8] of cell completely, and does not propagate it.
  - IPP version 2 will use it for implementing “subchannels”. See [Turner-96b] for details.
  - WUGS also does not use, and leaves undefined, all fields marked with diagonal lines (on following pages). IPP version 2 will use them.

External Control Cell Formats



External Control Cell CP to Switch

External Control Cell Switch to CP

- Operation Code (OPC) specifies operation.
- Control Offset (COF) identifies target.
- Return Value (RVAL) for returning status.
- Field or table entry to be accessed (FIELD).
- BI, RC, D, CYC, CS fields for each of three hops through switch.
- External address (EADR1,2,3) specifies internal addresses used in each of three hops.
- Return Header (RHDR) is cell header of returned cell.
- Information field (INFO) contains info read from/written to table entry/register field.
- Local Time (LT) field gives switch time at which information was accessed.
- Connection Management Data (CMDATA) used to correlate responses with requests.
- LINK_INFO only for IPP version 2.

Figure 21: External Control Cell Formats

Internal Data Cell Format

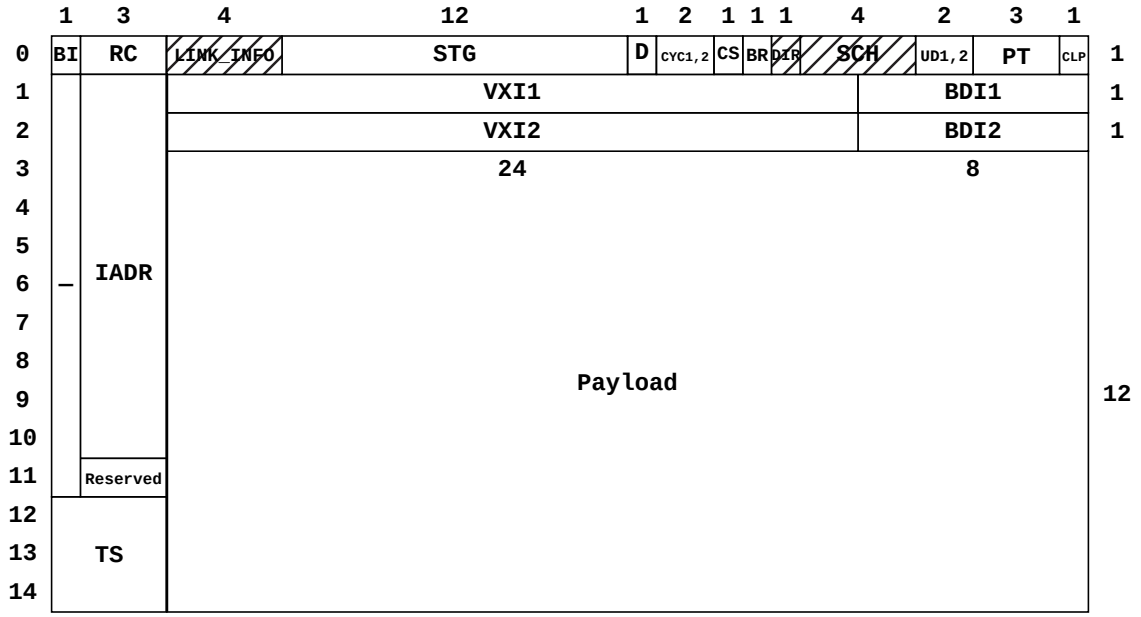


Figure 22: Internal Data Cell Format

- Busy/Idle (BI)
- Routing Control (RC)
- Internal Address (IADR)
- Time Stamp (TS)
- Virtual Path/Circuit Identifier (VXI1,2)
- Block Discard Index (BDI1,2) for packet level discarding
- Source (STG)
- Data (D), Recycling (CYC1,2), Continuous Stream (CS), By-pass Resequencer (BR), Echo Suppression/Upstream Discard (UD1,2), Payload Type (PT), Cell Loss Priority (CLP)
- Only for IPP version 2: LINK_INFO, DIR, SCH

Other Cell Formats

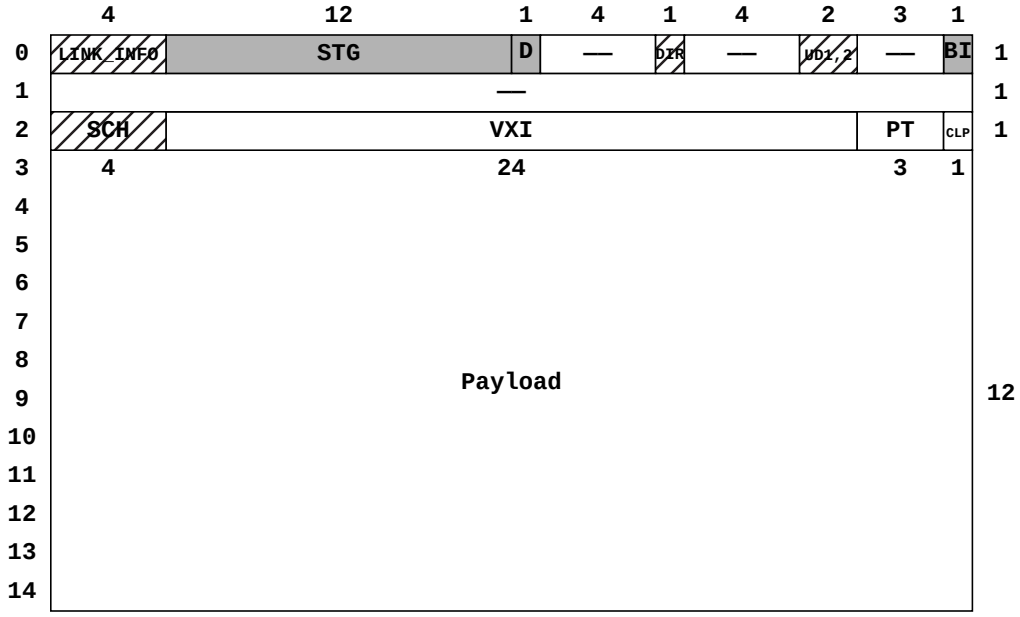


Figure 23: I/O and Recycling Data Cell Format

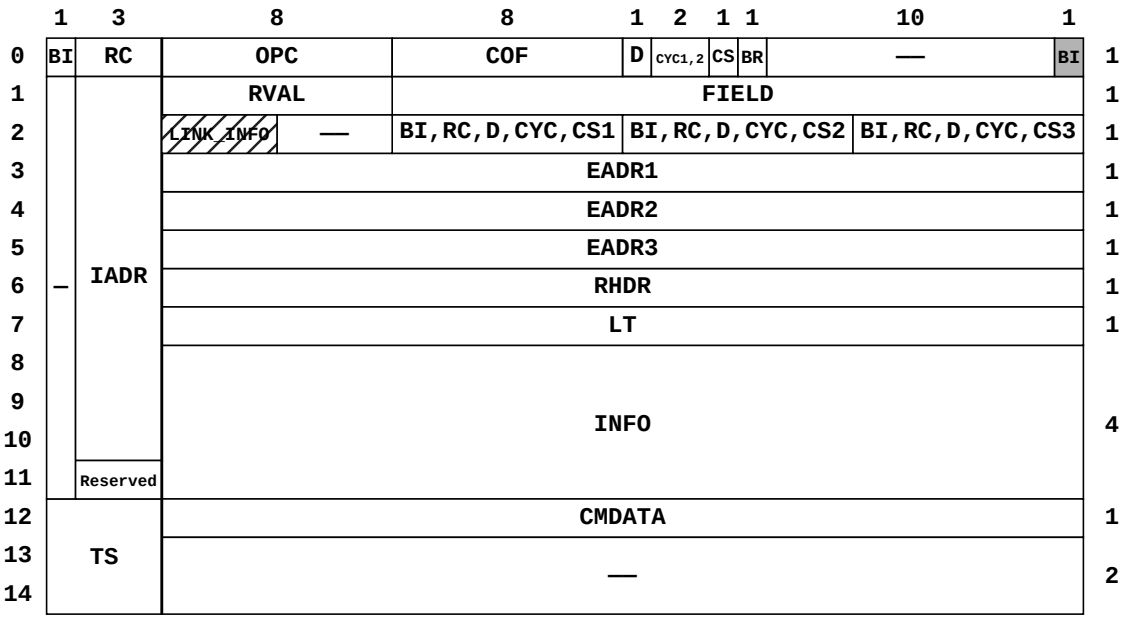


Figure 24: Internal Control Cell Format

- All shaded fields are only defined for cells in the recycling path.
- All fields marked with diagonal lines only used by IPP version 2.

Connection Setup Outline

- General recommendations on filling in fields of VXT entries
- Unicast connections
  - Setup, Teardown
  - Modify properties, Virtual Paths & Circuits
- One-to-many multicast connections
  - Adding & Removing Receivers, “Batch” setup, Modify properties
- Many-to-many multicast connections
  - Adding & Removing Senders, Adding & Removing Receivers, “Batch” set-up, Modify properties
- Notes on the dual OC-3 line card
- Monitoring traffic and error conditions in the switch

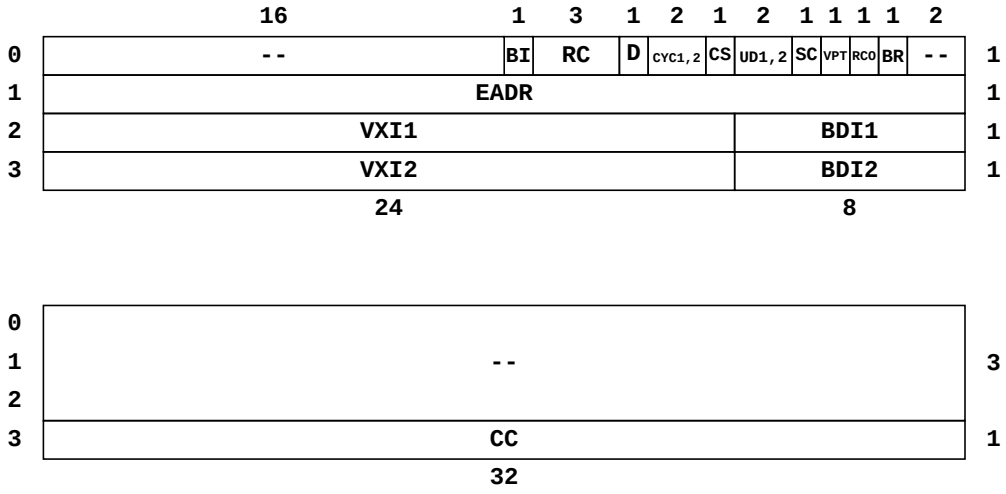


Figure 25: INFO Field Formats Used to Read/Write VXT Entries

Recommendations on Filling VXT Entries

- Virtual circuits require setting up the virtual path entry, too.
  - The virtual path entry must have virtual path termination (VPT) equal to 1. In this case, the rest of the virtual path entry's fields are completely ignored by the switch. VPT is ignored for virtual circuit entries.
- All VXT entries contain two sets of the fields: PORT (half of EADR, see Figure 19), CYC, VXI, UD, BDI

RC (binary)	Set 1	Set 2	Comments
000	used	ignored	EADR has unusual format for specific path cells
010	used	ignored	single copy goes to PORT1
001	ignored	used	single copy goes to PORT2
011	used for copy 1	used for copy 2	one copy to PORT1 and one copy to PORT2 (even if PORT1=PORT2)
1**	used for all copies	ignored except for PORT2	switch fabric sends copy to ports PORT1 up to and including PORT2 (requires PORT1 ≤ PORT2)

Table 26: How Routing Control (RC) Selects Among Two Sets of Forwarding Info

- Set D=1, unless you want to figure out why control cells are being created left and right, and your data is disappearing.
  - We should have left D out of the VXT entry (it is not in IPP version 2).
    - Someone may figure out a hack where they intentionally mutate data cells to control cells by this mechanism, for a useful purpose. This could be done with a specified periodic rate by sending a data cell through an infinite loop and copying off a mutated control cell in each pass through an IPP.
- Make recycling cells only (RCO) 1 for all entries except those where you want to allow cells arriving from the link to use the entry.
  - Multicast connection trees would still work if RCO=0 for all entries, but then cells from link could "spoof" into a recycling port.

More Recommendations on Filling Entries

- Continuous stream (CS)
  - Primarily for choosing whether cell goes into high or low priority FIFO while waiting to be transmitted on link. See “OPP XMB Details” on page 33.
  - Also used for discarding low priority cells if input port receive buffer becomes congested. See “IPP VXT (Virtual Circuit Translation Table) Function” on page 14.
- Bypass resequencer (BR)
  - Cells in connection experience reduced latency through output port resequencer, but may get out of order unless you are careful. See “OPP RSQ (Resequencer)” on page 24.
- Set CLP (SC)
  - If 1, all cells using entry will have CLP set to 1 (low priority). Useful if you don’t trust sender to restrict themselves to low priority.
- Some fields are only used by the switch for link bound cells. Their values are unimportant for cells that recycle.
  - BDI1,2: Only useful to make this non-0 for connections containing a sequence of non-interleaved AAL5 frames, possibly mingled with ATM Resource Management or OAM cells (see “OPP BDC Details” on page 29).
    - Uses early packet discard with hysteresis (EPDH) for discrete stream connections (CS=0) and partial packet discard (PPD) for continuous stream connections (CS=1).
    - Should be a different non-0 value for every distinct link bound connection passing out of the same OPP chip. Need not be the same value for all “leaves” in a multicast connection.
  - UD1,2: Echo suppression, a.k.a. “upstream discard”. Although originally motivated by many-to-many multicast connections, you may want to turn these bits on for all connections, except those connection tree “leaves” where you explicitly want data to echo back to the sending port (see “OPP RFMT (Reformatter)” on page 22).

Unicast Virtual Circuits

- Setup
  - See “Procedure to Establish Virtual Circuit” on page 59.
- Teardown
  - Same as setup except BI=0, and then all other VC entry fields are “don’t care”.
- Modify properties
  - Any or all of the fields in the VXT entry can be modified atomically with a single write control cell. The cell count will not be affected.

Virtual Paths

- See [Turn98, p. 14, “Virtual Paths and Circuits”] for basic idea of Virtual Paths.
- Same as virtual circuits, except:
  - VPT field of entry must be 0
  - all VCI values are “don’t cares”
  - opcode is WRVPXT
  - 24-bit FIELD of control cell must have VPI in most significant 8 bits.
- Teardown
  - VPT must be 0 for a virtual path entry, and BI must be 0 for VXT to discard all cells with that VPI.

Procedure to Establish Virtual Circuit

```
-- Here is a Jammer command line that has the same effect
-- as the procedure below.  All occurrences of “d” indicate
-- a “don’t care” value, meaning that it will be ignored by
-- the switch hardware.
--
-- write vcxt inPort inVCI 1 2 1 0 d cs ud d sc d 0 br outVPI out-
VCI bdi d d d outPort d

-- The general form of this command is:
--
-- write vcxt targetPort inVCI BI RC D CYC1 CYC2 CS UD1 UD2 SC VPT
RCO BR VPI1 VCI1 BDI1 VPI2 VCI2 BDI2 PORT1 PORT2

procedure setup_unicast_vc (inPort, inVPI, inVCI,
                            outPort, outVPI, outVCI,
                            ud, bdi, cs, sc, br)
begin
   -- Assume that inVPI and inVCI are in range
   -- (see “IPP VXT Details: Data Cell Processing” on page 16).
   -- Assume that VP entry already has VPT=1.
   -- Assume that VC entry inVCI in inPort is currently unused.

   entry.BI := 1;
   entry.RC := 010;  -- 001 would work as well if set 2 is defined
   entry.PORT1 := outPort;
   entry.CYC1 := 0;
   entry.VXI1 := (outVPI concatenated with outVCI);
   entry.{UD1,BDI1} := {ud,bdi};
   entry.{PORT2,CYC2,VXI2,UD2,BDI2} := “don’t care”;
   entry.{CS,SC,BR} := {cs,sc,br};
   entry.VPT := “don’t care”;
   entry.RCO := 0;  -- otherwise cells from link will be discarded
   entry.D := 1;

   -- for sendControlCellToIPP definition, see page 70
   sendControlCellToIPP (targetport => inPort,
                  opc => WRVCXT, field => inVCI, info => entry);

   -- Verify INFO of return control cell, if desired. It should be
   -- the same as sent out, at least in bit positions not marked
   -- “--” in Figure 25).
end;
```

This page should be replaced with [Turn98, p. 17, “Multicast Connection Trees”]

This page should be replaced with [Turn98, p. 18, “Adding an Endpoint”]

This page should be replaced with [Turn98, p. 19, “Dropping an Endpoint”]

One-to-Many Multicast Connections

- See [Turn98, p. 17, “Multicast Connection Trees”] for general idea.
- Adding Receiver
  - See [Turn98, p. 18, “Adding an Endpoint”] for basic idea.
  - Control cells and table entry contents are the same as unicast, except
    - RC=011
    - both sets of PORT, CYC, VXI, etc. should be filled in
    - CYC bits need to be 1 for copies that recycle
    - RCO should be 1 for all but the root of the multicast tree.
- Removing Receiver
  - See [Turn98, p. 19, “Dropping an Endpoint”]
  - To guarantee FIFO cell ordering within the connection, must use transitional time stamping when modifying the one table entry necessary. See [Turn89, p. 23, “Avoiding Misordering During Transitions”].
    - Use opcode WRVPXTTR for virtual paths and WRVCXTTR for virtual circuits.
- “Batch” setup
  - For signaling software, probably simplest to implement add receiver and remove receiver operations, and create connections with many receivers by adding them sequentially. With WUGS-20, the software and hardware “work” required is at most twice that of creating the connection with all desired receivers from the start.
  - It would not be difficult to implement an operation to create a connection that initially has many receivers.
- Modify properties
  - Any or all of the fields in the VXT entry can be modified atomically with a single control cell. The cell count will not be affected.

This page should be replaced with [Turn98, p. 20, “Scalable Many-to-Many Multicast”]

Many-to-Many Multicast Connections

- See [Turn98, p. 20, “Scalable Many-to-Many Multicast”] for general idea.
- Fully flexible - the senders can be chosen completely independently of the receivers.
- Many-to-many connections can be implemented similarly to one-to-many connections
  - The main difference is that all senders have a VXT entry that forwards and recycles a single copy of all cells to a common VXT entry that is the root of the one-to-many multicast tree.
- Adding Sender
  - Just set up point-to-point forwarding entry from sender’s port to root of one-to-many multicast tree.
- Removing Sender
  - Disable sender’s entry.
- “Batch” setup, Adding and Removing Receivers, and Modify properties
  - Similar to the corresponding operations in one-to-many connections

Suppressing Echo Cells in Many-to-Many Connections

- Without the echo suppression feature (a.k.a. upstream discard), setting up a scalable many-to-many connection spanning multiple WUGS-20 switches would cause every cell sent in to loop forever.
- With echo suppression, the cell is sent to all outputs ports except for the one where it arrived.
- The maintenance register fields used to identify “same ports” for this function are called “Trunk Group Identifiers” rather than “Port Identifiers” to emphasize that they can be made the same for multiple ports
  - Useful if there are two switches with multiple fiber pairs between them, and signaling software allows a many-to-many multicast connection to use different ports for the two directions of traffic.
- We designed system to allow possibility of getting back echo cells, in case it was ever desired.
  - Do this by making field UD in the last VXT entry before the cells leave the switch equal to 0.
  - Example 1: sending a unicast connection from subport A of a dual OC-3 line card out to subport B of the same line card. As far as the switch is concerned, this is the same port because it is the same IPP/OPP pair.
    - Note that this implies that the echo suppression feature is not general enough to allow multiple subports to be in a many-to-many multicast connection with echo suppression.
  - Example 2: a host in a many-to-many connection requests to receive copies of whatever it sends.

Dual OC-3 Line Cards

- The line cards have markings to distinguish “subports” A and B, and so does the front panel of the switch.
  - These are both connected to the same IPP/OPP pair.
- **Control cells cannot be sent to switch from subport B!**
  - Cell must have VPI=0, VCI=32 **when it enters an IPP from the link** to be recognized as a control cell. If a cell is translated to VPI=0, VCI=32 and re-cycled, that does not make it become a control cell.
- The line card replaces VPI[7] of all **cells received from the fiber** with a subport identifier, 0 for A and 1 for B, before passing the cell on to the IPP chip.
  - Cells arriving on subport A with VPIs in range [0,127] access VXT entries [0,127] for their virtual path lookup.
  - Cells arriving on subport B with VPIs in range [0,127] access VXT entries [128,255] for their virtual path lookup.
  - If you wish to have connections with VPI=x on subport B, MREG.VPCount must be at least (128+x), or such cells will be discarded for having a VPI that is too large. See “IPP VXT Details: Data Cell Processing” on page 16.
  - Cells arriving on either subport with VPIs in range [128,255] will be discarded by the IPP for having a bad HEC (counted in both MREG.BadHEC-Counter and MREG.ReceiveCellCounter), if the cell’s HEC on the fiber was correct.
- The line card uses VPI[7] of cells transmitted by the **OPP** to select a subport, and replaces VPI[7] of all cells with 0.
  - The HEC is recalculated by the line card, ignoring the HEC calculated by the switch.
- For further details, see the Link Spec Document [RF-94a, Section 6.2]

Monitoring Traffic

- To measure data rates at any point the switch maintains a counter, send at least two control cells reading the counter, and use the difference in the counters and the local times (LT) in the returning control cells to calculate a data rate.

```
cells = (cell count 2) - (cell count 1);
duration_celltimes = (local time 2) - (local time 1);
duration_sec = (duration_celltimes * 16)
                                / switch_clock_in_hz;
cells_per_sec = cells / duration_sec;
bits_per_sec = (cells * 53 * 8) / duration_sec;
```

  - Counters wrap around from max value to 0.
  - All counters can be written, but there is no reason to do so using the method above.

- Statistics available
  - Per VP/VC cell count
  - Normal traffic
    - Cells received by IPP from link adaptor card (IPP ReceiveCellCounter) and sent to link from OPP (OPP TransmitCellCounter)
    - Cells received by IPP from OPP on recycling path (IPP or OPP Recycling-CellCounter)
  - Errors
    - Cells discarded by IPP due to incorrect HEC (IPP BadHECCounter)
  - Congestion
    - (IPP VXTCS0DiscardCounter)
    - (IPP RCBCLP0OverflowCounter and RCBCLP1OverflowCounter)
    - (IPP CYCBDiscardCounter)
    - (OPP XMBCS0OverflowCounter and XMBCS1OverflowCounter)
    - (OPP TooLateDiscardCounter)
    - (OPP ResequencerOverflowCounter)

Monitoring Error Conditions

- Can read the desired maintenance register fields individually for every IPP and OPP chip periodically.
- Also possible to monitor all IPPs (or all OPPs) for many error conditions by sending single control cell periodically.
  - Send control cell exactly like the following call would do, except make RC1=111 and fill in EADR1 with range PORT1=0 up to PORT2=7.

```
-- for sendControlCellToIPP definition, see page 70
sendControlCellToSwitch (targetPort,
                cof => 1,       -- 1 for IPP, 0 for OPP
                opc => ERRORS,  -- [SAD, Figure 22]
                field => 0,
                info => “don’t care”,
                portBackToCP)
```

  - The control cell will be copied to all IPPs (or OPPs), and only those with MREG.ReportErrors (or OPP MREG.ReportErrors) equal to 1 and at least one of the following error flags on (i.e., 1) will send back a response. All others will discard the control cell quietly.
    - For IPPs: HardwareReset, LinkWasDisabled, ParityError, VXIOutOfRange, BadControlCell, BadATMSignalingCell
    - For OPPs: HardwareReset, ParityErrorPlane3, ParityErrorPlane2, ParityErrorPlane1, ParityErrorPlane0
  - Response cells contain TrunkGroupIdentifier as well as error flag values, so monitoring software can narrow down the possible ports that have errors.
- Error flags can be written as a group, or individually.
  - Group writing may inadvertently turn off an error flag that turned on since the last read control cell. Use FIELD values of individual error flags when turning them off.
- For software, could be easier to treat such response cells as spontaneously generated interrupts
  - as opposed to responses to the ERRORS control cells, because the number of responses is unknown when the ERRORS control cell is sent out.

Pseudocode for Sending Control Cell

- This handles the most common control cell type that makes two passes through the switch fabric.
  - Three passes are only needed for testing certain data paths in a switch with a multistage switching fabric.

```
procedure sendControlCellToSwitch (targetPort, cof,
                                   opc, field, info, portBackToCP)
begin
   cell.{OPC,FIELD,INFO} := {opc,field,info};
   cell.COF := cof;  -- 0 for OPP, 1 for IPP;
   cell.RVAL := 1;   -- NOT_PROCESSED (see [SAD, Figure 25])
   cell.BR := 1;     -- Unfortunately, not filled in properly
                     -- for control cells. See [SAD, Section 11].
   cell.CMDATA := “unique value, to identify response cell”;
   -- go to target port on first pass through switch
   cell.BI1 := 1;
   cell.RC1 := 010;   -- 010 arbitrarily chosen over 001
   cell.D1 := 0;
   cell.CYC1 := 1d;   -- “d” for “don’t care”
   cell.CS1 := 1;
   cell.EADR1 := {PORT1 => targetPort, PORT2 => “don’t care”};
   -- go back to CP on second and final pass through switch
   cell.BI2 := 1;
   cell.RC2 := 010;
   cell.D2 := 0;
   cell.CYC2 := 0d;   -- “d” for “don’t care”
   cell.CS2 := 1;
   cell.EADR2 := {PORT1 => portBackToCP, PORT2 => “don’t care”};
   -- No third pass through switch
   cell.{BI3,RC3,D3,CYC3,CS3,EADR3} := “don’t care”;

   repeat
      send the cell with VPI/VCI so that it arrives to the switch
         with VPI=0, VCI=32, directly or through another switch.
      give it some time to come back;
   until (response cell comes back);
   -- Verify as many of the following fields as you want, which
   -- should have values indicated: OPC (no change), COF (cof-2),
   -- RVAL (SUCCESS=0), FIELD (no change), RHDR (no change)
end;
```

Needs title

```
-- sendControlCellToOPP is same, except cof is 0.

procedure sendControlCellToIPP (targetPort, opc, field, info)
begin
   sendControlCellToSwitch (targetPort, cof => 1,
                            opc, field, info, portBackToCP);
end;
```

Guide to WUGS Operational Scenarios

- Scenario 1 - control cell writing a virtual path VXT entry
- Scenario 2 - data cell using that unicast virtual path connection
- Scenario 3 - data cell in many-to-many virtual circuit connection
- Scenario 4 - no op control cell that makes three passes, to test a particular path through switch fabric
  - This scenario is given using a 16 port “WUGS-40” switch fabric, since that makes switch fabric scenarios more interesting than 8 port.

Notes on All Scenarios

- The flow proceeds in steps, labeled with circled numbers.
- Any cell fields that have changed since the previous step are shaded.

Scenario 1, p. 1 of 2

- Control cell writing a virtual path VXT entry
  - First pass, through IPP 5 (connected to control processor) and OPP 1.
- **1**: control cell arrives from link to IPP 5 with VPI=0, VCI=32, in CP to switch external control cell format.
- **2**: no change
- **3**: First of three sets of routing info used to fill in BI, RC, D, CYC, CS, and IADR of cell (D,CYC,CS abbreviated “DCC” in figure). Other two sets shifted forward. Time stamp (TS) filled in by RFMT. RFMT converted cell from external to internal control cell format.
- **4**: switch fabric has used and modified IADR while routing cell to OPP 1. RFMT selects set 1 of CYC1,2 because RC /= 001. Cell has CYC1=1, so it will recycle.
- **5**: OPP MREG has seen arriving COF=1 is not 0, so no operation performed there. COF decremented.

Control Cell Processing

Description:

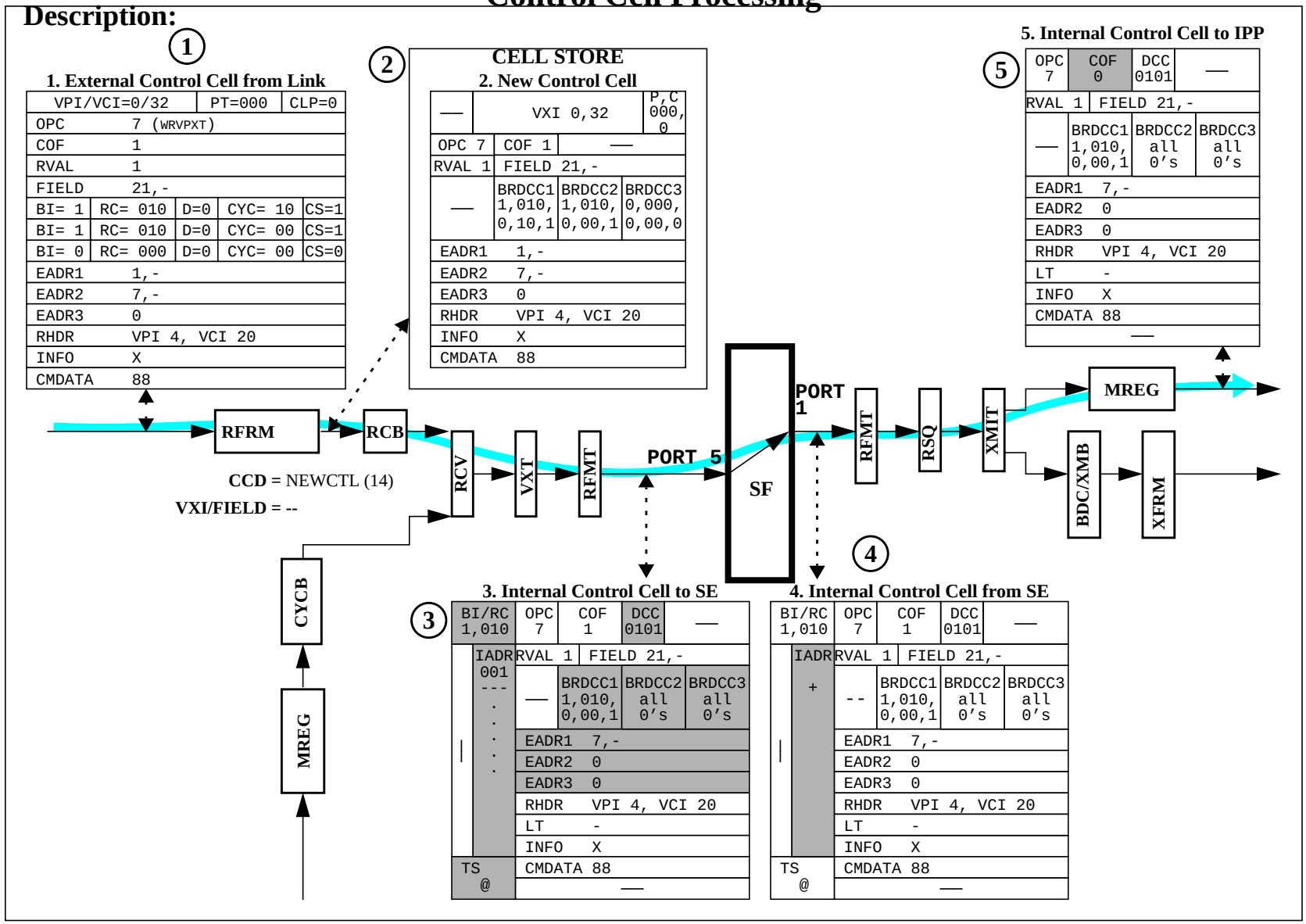


Figure 27: Scenario 1, p. 1 of 2

Scenario 1, p. 2 of 2

- Control cell writing a virtual path VXT entry
  - Second pass, through IPP 1 (where write op performed) and OPP 7 (leading back to control processor).
- **1**: control cell arrives from recycling path to IPP 1 in internal control cell format. It has COF=0 and OPC=”write VP entry” (WRVPXT), so MREG passes it on with an internal code indicating that VXT should perform the operation.
- **2**: MREG also decremented COF, wrapping around to FF hex.
- **3**: VXT has written virtual path entry 21 specified by FIELD, and read back the value written. RFMT put in return value (RVAL) of SUCCESS (0), time of operation in LT, and value read back from VXT into INFO.
  - As in first pass, first of three sets of routing info used to fill in BI, RC, D, CYC, CS, and IADR of cell. Other two sets shifted forward. Time stamp (TS) filled in by RFMT.
- **4**: switch fabric has used and modified IADR while routing cell to OPP 7. Cell has CYC1=0, so it will go to link and thus RFMT converts it from internal to external format, using RHDR field to fill in ATM cell header.
  - It has CS=1, which chooses CS=1 FIFO in XMB. All control cells avoid the EPDH/PPD features of BDC (i.e., treated as if they have BDI=0).
- **5**: OPP XFRM transmits cell to link.

Control Cell Processing

Description:

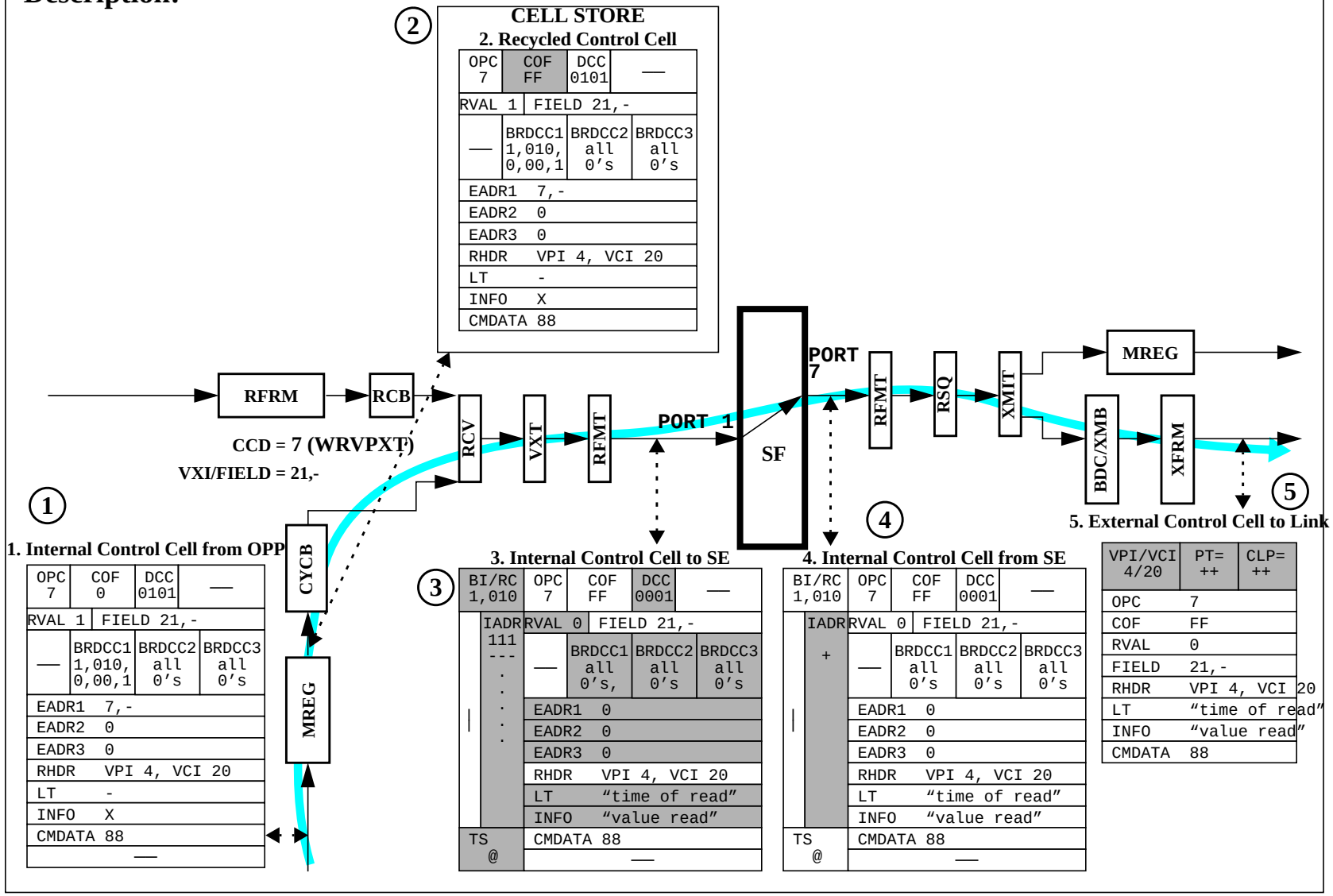


Figure 28: Scenario 1, p. 2 of 2

Scenario 2, p. 1 of 1

- Data cell in unicast virtual path connection
- **1**: data cell arrives from link to IPP 1 in external data cell format with VPI=21, VCI=15. **2** has no change.
- **3**: VP entry 21 in the VXT is accessed. It has VPT=0, so this is a virtual path connection, and no VC entry is accessed.
- **4**: Since this is a virtual path, the VCI was not modified, only the VPI. VXT set CLP to 1 because SC=1. Time stamp (TS) filled in by RFMT. RFMT converted cell from IO/recycling to internal data cell format, and filled in source trunk group (STG) with 1.
  - Assume all ports have Trunk Group Identifier equal to their port numbers.
- **5**: switch fabric has used and modified IADR while routing cell to OPP 2. Cell has CYC1=0, so it will go to the link.
  - Cell's source trunk group (STG)=1 is not the same as the OPP's Trunk Group Identifier (TGI)=2, so this is not an echo cell, and it is not discarded as such.
- **6**: RFMT converted it from internal to IO/recycling format.
  - Since BDI=16 is not 0, cell will be subject to early packet discard with hysteresis (EPDH) state machine in BDC (EPDH because CS=0), and if not discarded will go into CS=0 FIFO (low priority) in XMB.
- **7**: OPP XFRM transmits cell to link. Congestion (C) bit in PT set if XMB congested.

Data Cell Processing

Description:

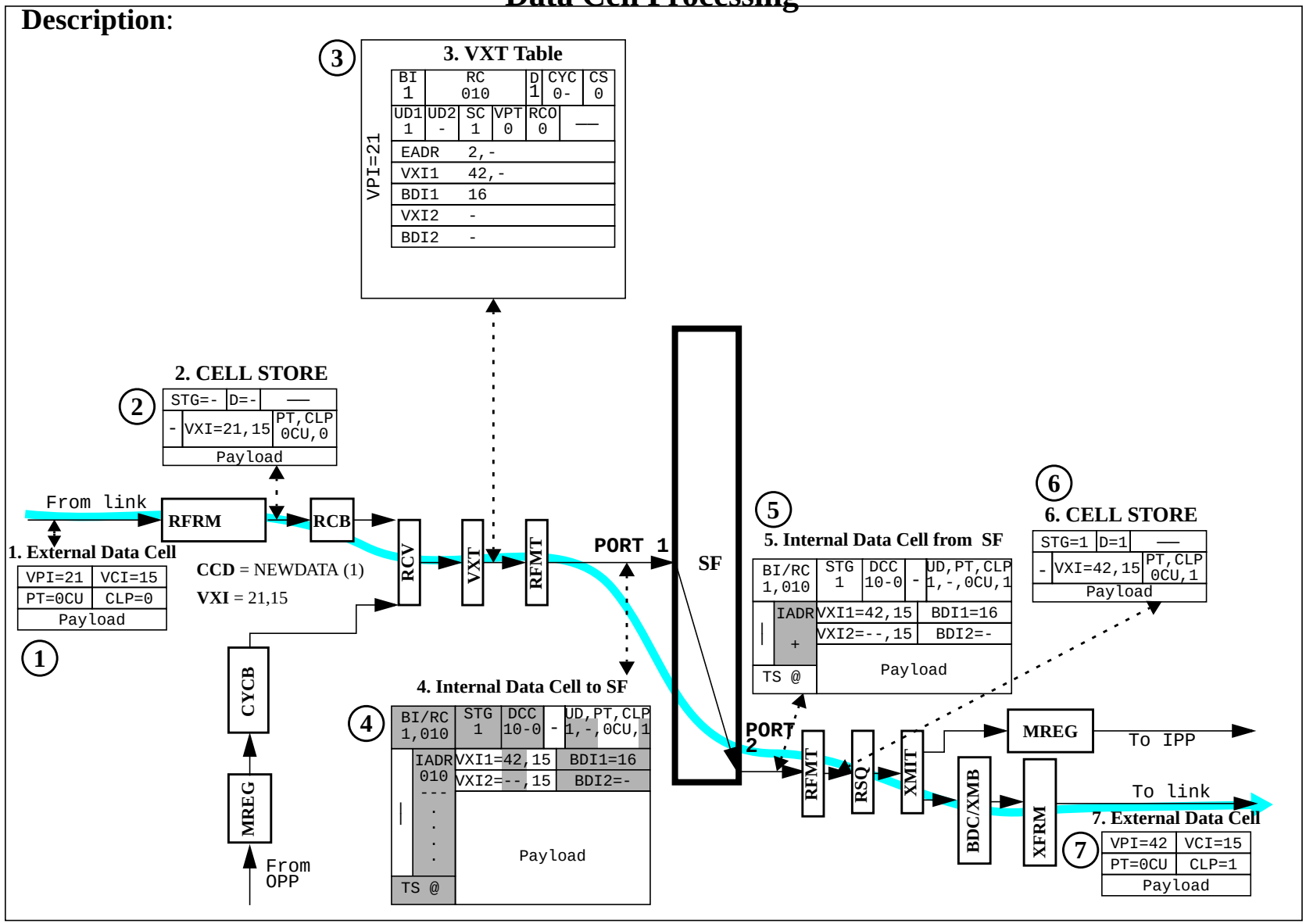


Figure 29: Scenario 2, p. 1 of 1

Scenario 3, p. 1 of 2

- Data cell in many-to-many virtual circuit

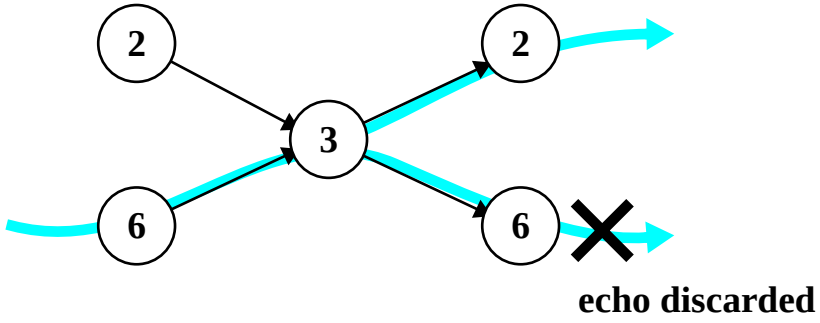


Figure 30: Many-to-Many Connection Tree for Scenario 3

  - First pass, through IPP 6 and OPP 3
- Only significant differences from Scenario 2 are mentioned.
- **3**: VP entry 255 in the VXT is accessed. It has VPT=1, so this is a virtual circuit connection, and VC entry 8 is accessed as well. Its fields will be used.
- **4**: Both VPI and VCI have been translated. RFMT filled in source trunk group with 6.
- **5**: switch fabric routed cell to OPP 3. Cell has CYC1=1, so it will recycle.
- **6**: RFMT converted format. Cell recycles, so it will not be subject to discard in BDC.
- **7**: MREG passes data cells unmodified.

Data Cell Processing

Description:

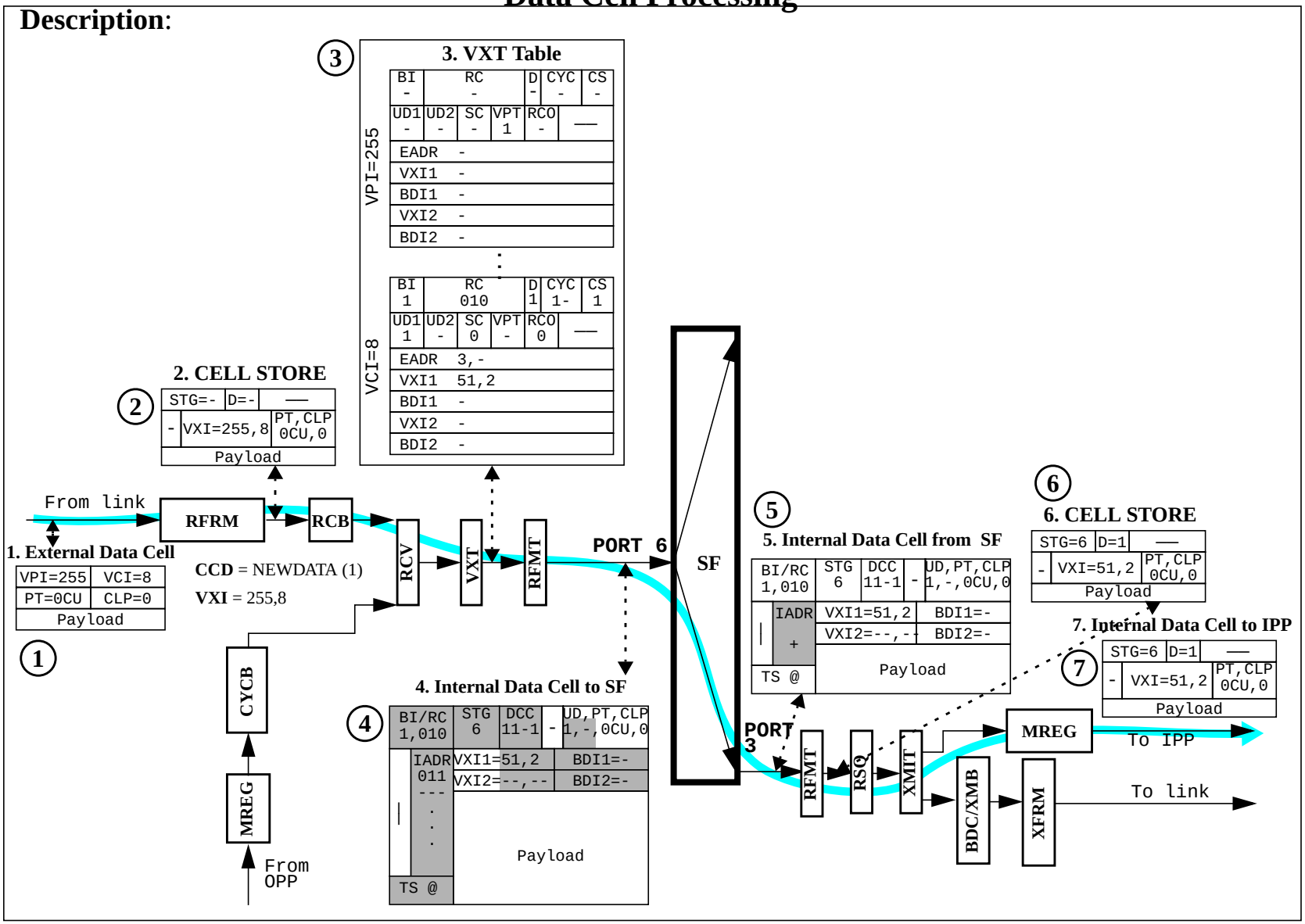


Figure 31: Scenario 3, p. 1 of 2

Scenario 3, p. 2 of 2

- Data cell in many-to-many virtual circuit connection
  - Second pass, through IPP 3 to both OPP 2 and 6. Copy to OPP 6 is discarded because it is an echo cell and echo suppression is turned on for connection.
- 1 & 2: MREG passes data cells unmodified.
- 3: Again, both VP and VC entries are accessed. Here two copies are indicated.
- 4: For a recycling data cell, the source trunk group is propagated unmodified, to preserve the value it was assigned when it first arrived to the switch.
- 5b: Not only did switch fabric use and modify IADR to route this copy of the cell to port 6, it also changed routing control (RC) from 011 to 001, so RFMT knows to use set 2 of CYC, VXI, UD, etc.
  - Cell is link bound (CYC2=0) and its STG=6 is the same as the OPP's TGI, so this is an echo cell. The connection has echo suppression on (UD2=1), so it is discarded by RFMT with no state updated.
- 5a: The switch fabric changed RC of this copy from 011 to 010. RFMT uses set 1 of CYC, VXI, UD, etc. Not an echo because OPP 2 has TGI=2.
- **6a**: RFMT converted it from internal to IO/recycling format.
  - Since BDI=22 is not 0, cell will be subject to partial packet discard (PPD) state machine in BDC (PPD because CS=1), and if not discarded, will go into CS=1 FIFO (high priority) in XMB.

Data Cell Processing

Description:

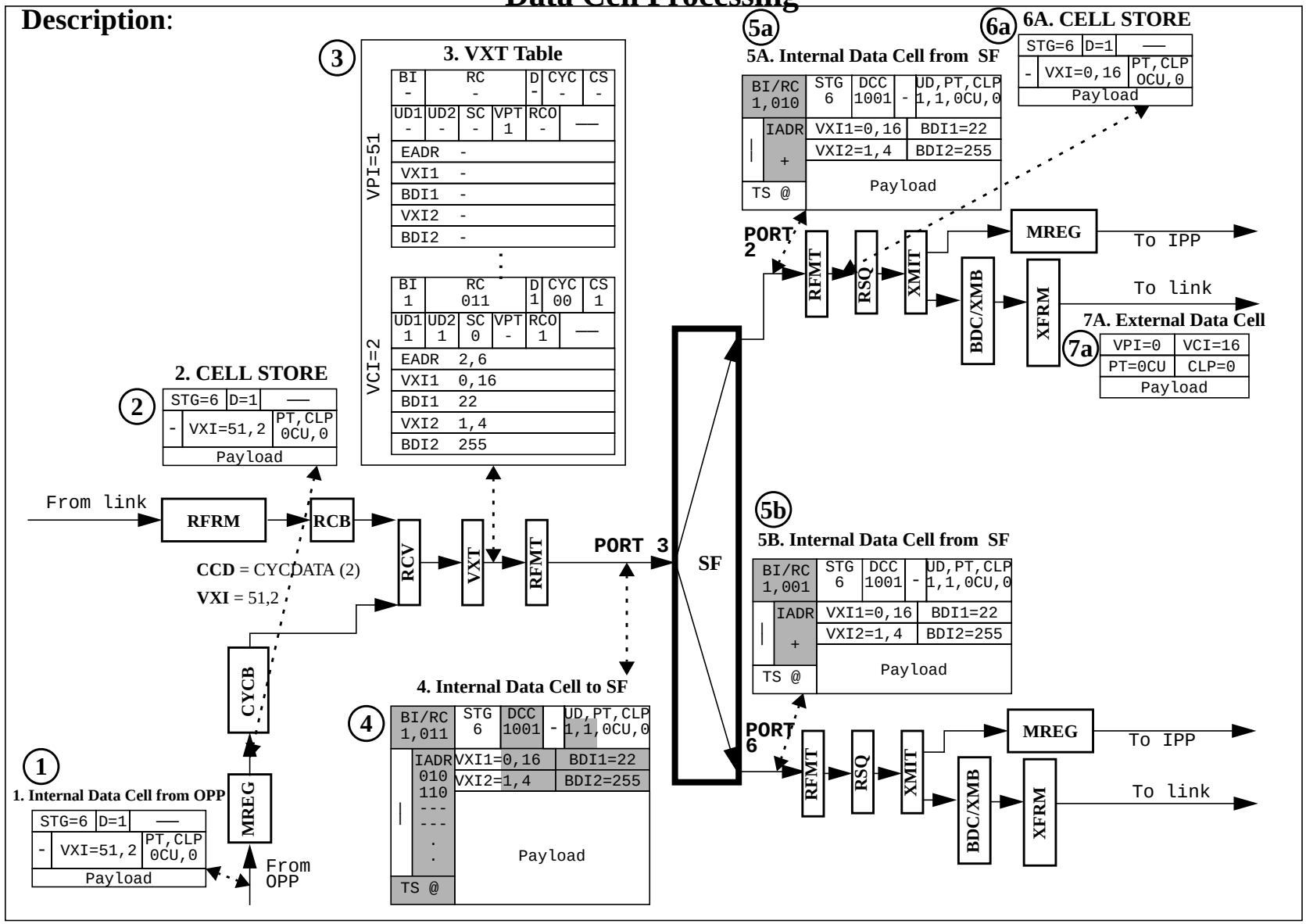


Figure 32: Scenario 3, p. 2 of 2

Scenario 4, p. 1 of 5

- No op control cell that makes three passes
  - First pass, through IPP 5 (connected to control processor) and OPP 2
- Very similar to Scenario 1, p. 1.
- 1: Differences here are OPC=no op, COF=3, FIELD is unused for no op control cells, and there are three passes worth of routing info in the cell that will be examined by switch, because the recycling bit is 1 for the first two passes.
- 3: Because this switch fabric has three stages, rather than the WUGS-20's one stage, EADR and IADR are formatted differently.
  - More rows of IADR are used by switch fabric. See p. 2 of this scenario for details of cell routing through switch fabric.
  - See [SAD, Figure 42] for details of IADR for switch fabrics with up to 9 stages.

Control Cell Processing

Description: NOP (Opcode 0) test cell (page 1 of 5)

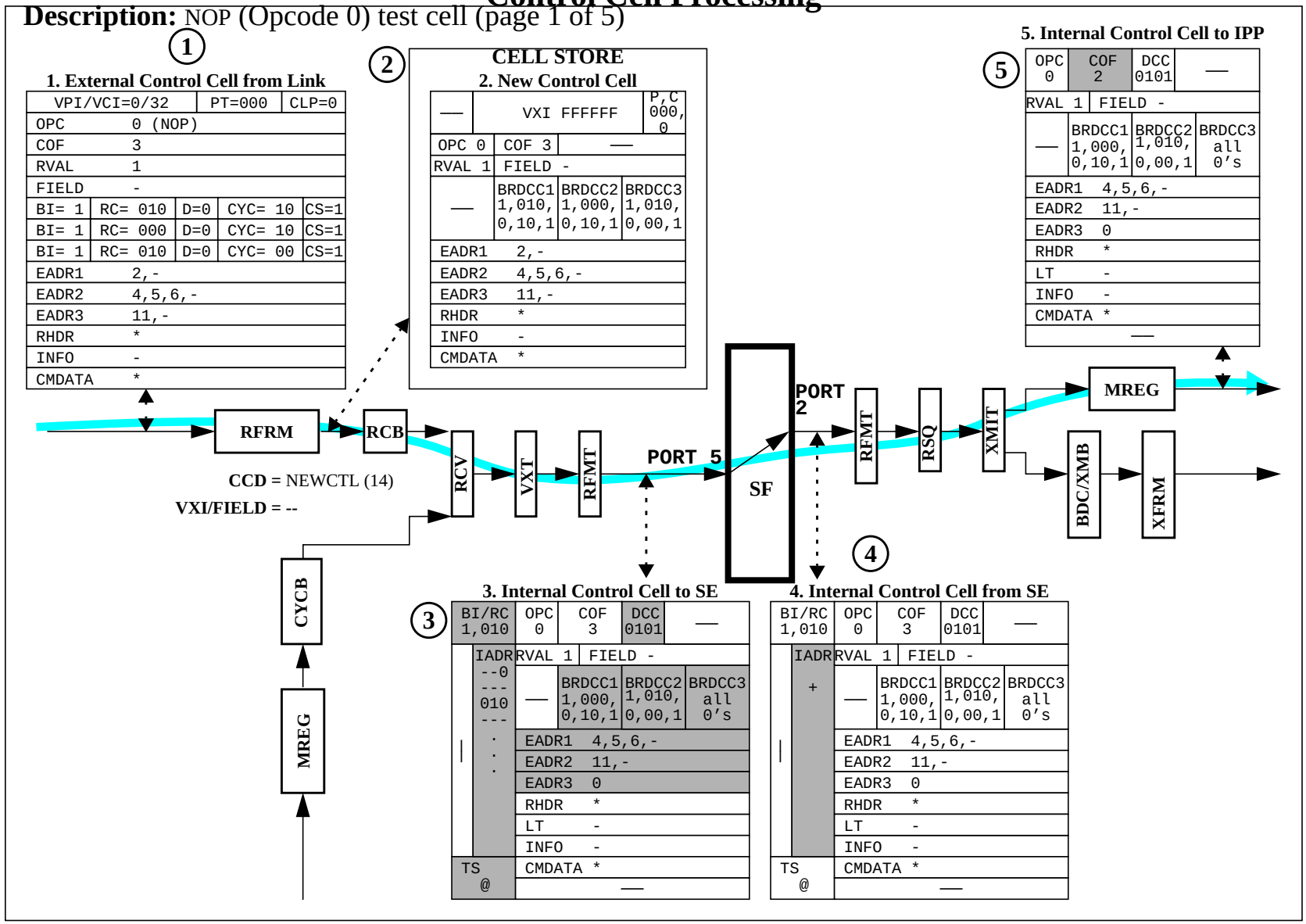


Figure 33: Scenario 4, p. 1 of 5

Scenario 4, p. 2 of 5

- No op control cell that makes three passes
  - First pass through switch fabric, demonstrating unicast cell in 16 port switch fabric (not implemented in WUGS-20)
- 1:
- 2: As mentioned in “SE Operational Details” on page 46, cells are distributed uniformly in the first half of the switch fabric using a counter.
  - This cell arrived at input port 5 of an SE when its counter was 2 (010 binary), so it is sent to output (5+2) mod 8 = 7.
  - Distribution stages do not modify the IADR field.
- 3: If the number of ports is not an integer power of 8, the middle stage is a hybrid of distribution and copying/routing. The effect here is to balance traffic across the 4 links available.
  - The most significant bit of the output port number 2 (0) is always used as the lsb of the SE output port in this middle stage.
  - The most significant 2 bits of the output port are determined by the current counter and input port, as in the first stage. counter+input = 101+110 = 011. The 2 lsb’s 11 are used.
  - IADR is shifted up by two rows to make the next bits of the address move to the top.
- 4: In the last stage, strict copying/routing is performed, and the counter is ignored.

Cell Processing

Description: Cell routing using RC=010. Cell arrives at port 5 and is destined to port 2. (page 2 of 5)

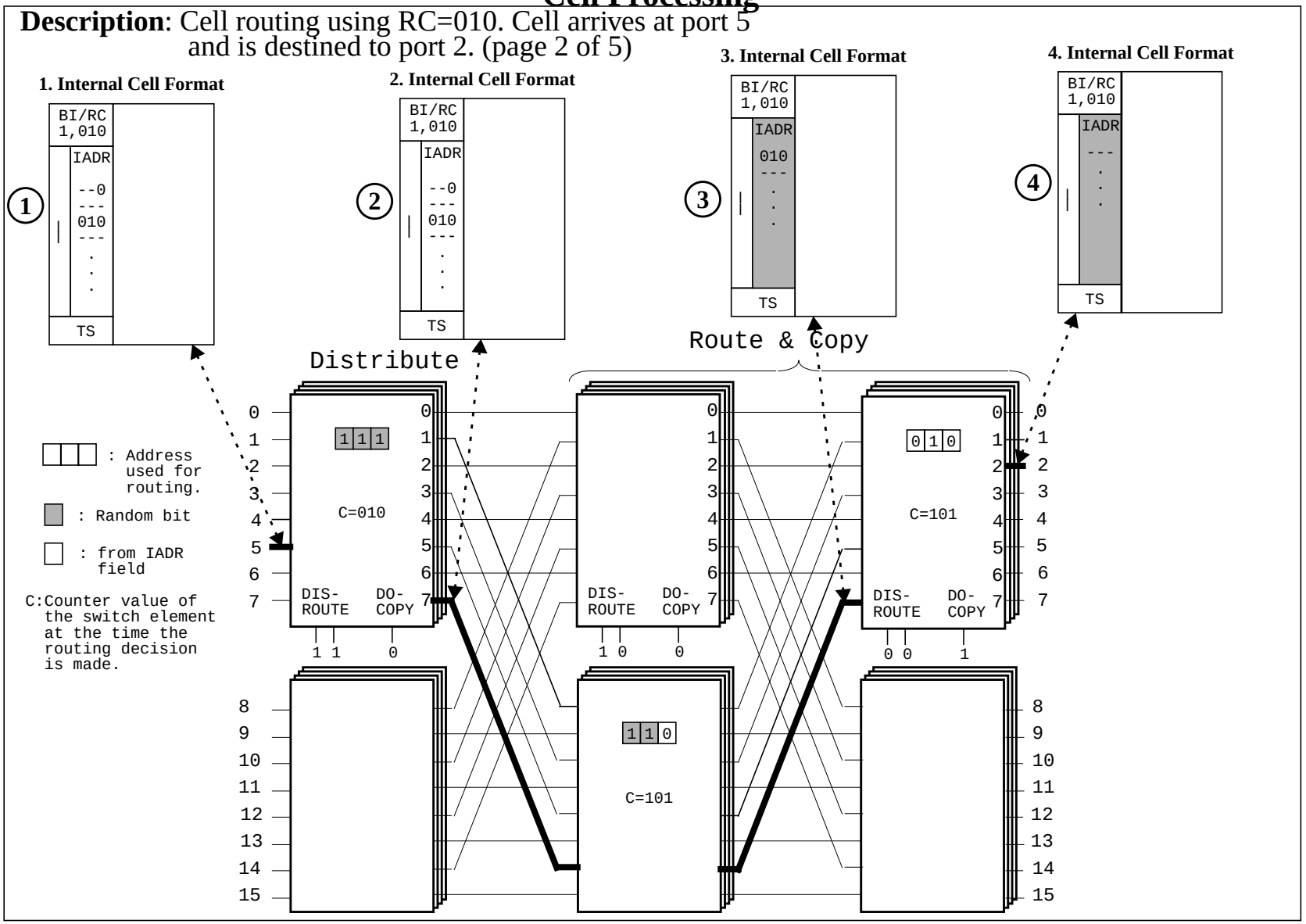


Figure 34: Scenario 4, p. 2 of 5

Scenario 4, p. 3 of 5

- No op control cell that makes three passes
  - Second pass, through IPP 2 and OPP 14
- Nothing surprising here. Very similar to previous control cell scenarioes.

Control Cell Processing

Description: NOP (Opcode 0) test cell (page 3 of 5)

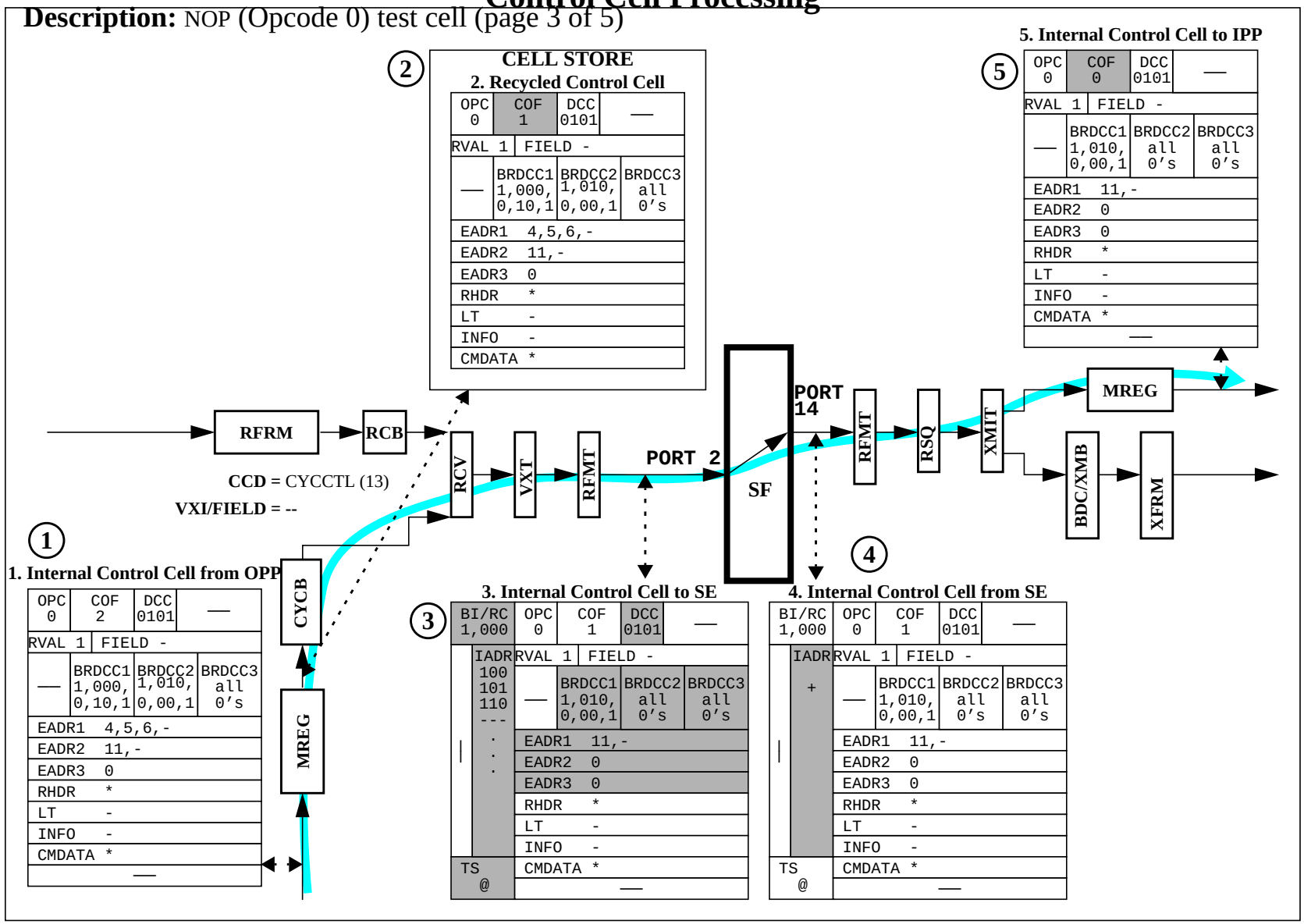


Figure 35: Scenario 4, p. 3 of 5

Scenario 4, p. 4 of 5

- No op control cell that makes three passes
  - Second pass through switch fabric, demonstrating specific path cell in 16 port switch fabric (not implemented in WUGS-20)
- Cells that use specific path routing (RC=000) have one SE output port number in each row of IADR.
- Every stage (whether it is normally a distribution, routing/copying, or “hybrid” stage) uses the top row to route the cell, and shifts IADR up by one row to prepare it for the next stage.
- In WUGS-20, specific path routing does not allow one to specify anything more precisely than the other routing options, because there are no distribution stages, only one routing/copying stage.

Cell Processing

Description: Cell routing using specific path routing (RC=000) (page 4 of 5)

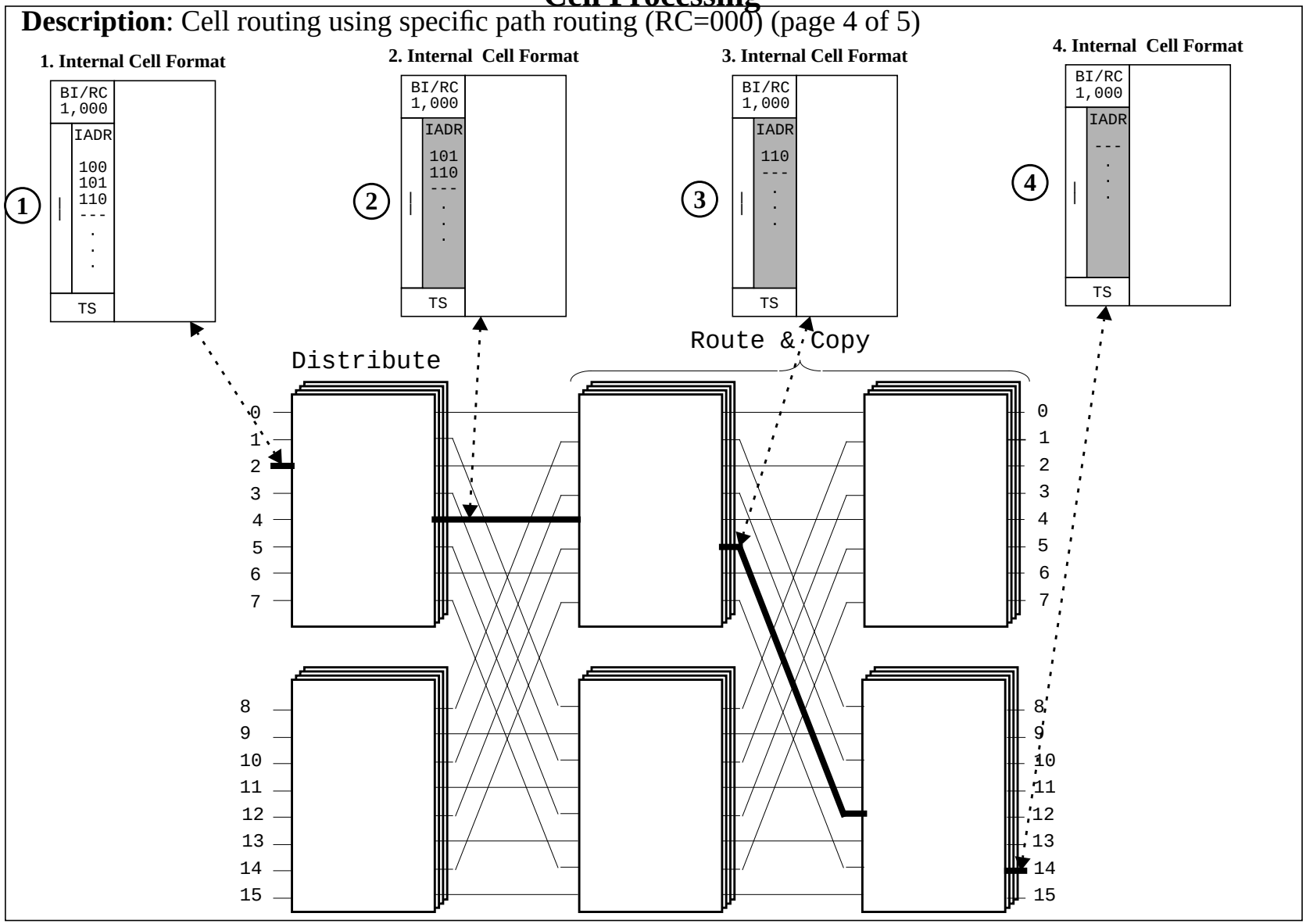


Figure 36: Scenario 4, p. 4 of 5

Scenario 4, p. 5 of 5

- No op control cell that makes three passes
  - Third and final pass, through IPP 14 (where no op is performed) and OPP 11 (leading back to control processor)
- Very similar to previous control cell scenarios.
- No op performed in MREG, because cell arrived to IPP with COF=0. Only effect is to modify return value (RVAL) and local time (LT) of control cell.

Control Cell Processing

Description: NOP (Opcode 0) test cell (page 5 of 5)

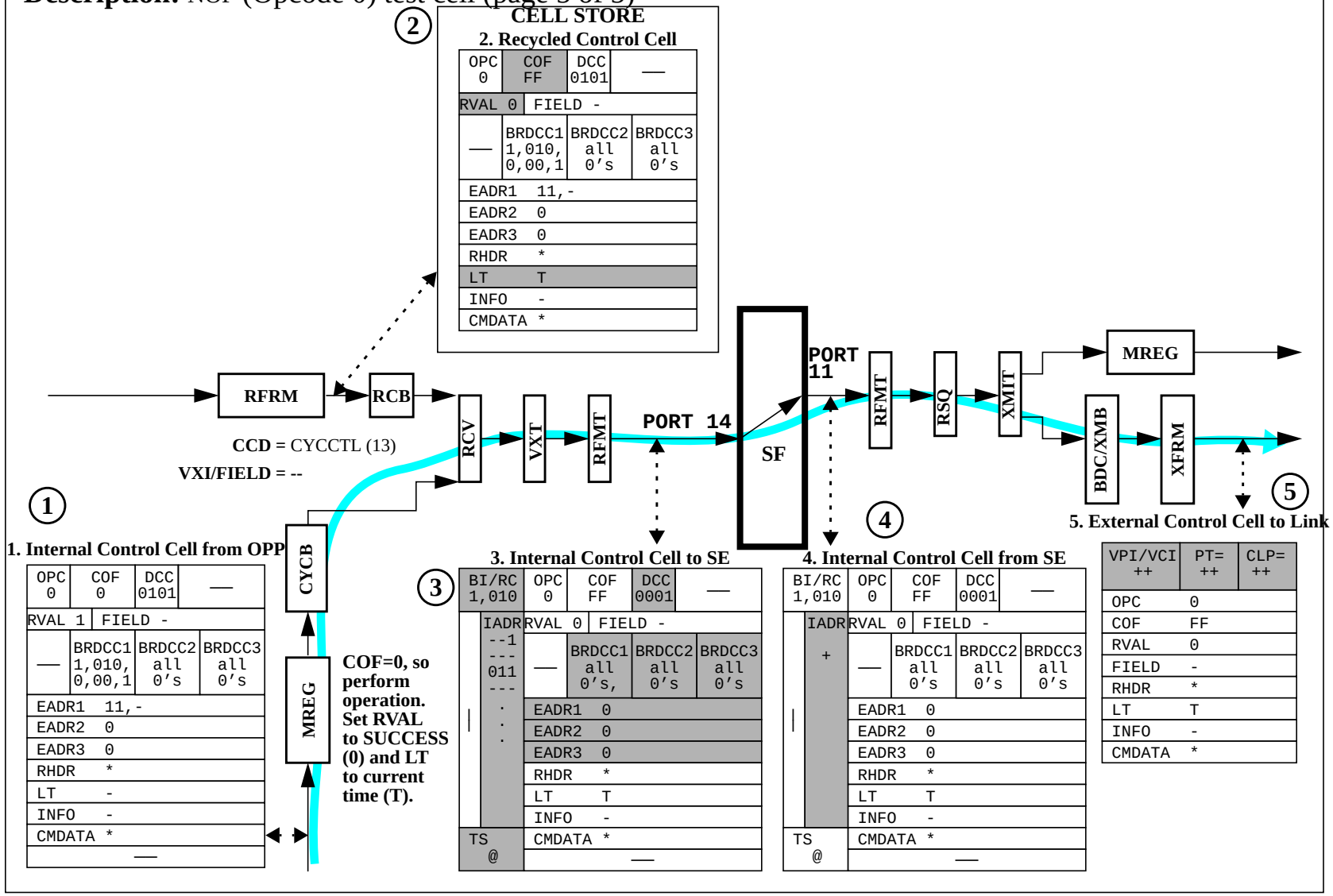


Figure 37: Scenario 4, p. 5 of 5

•

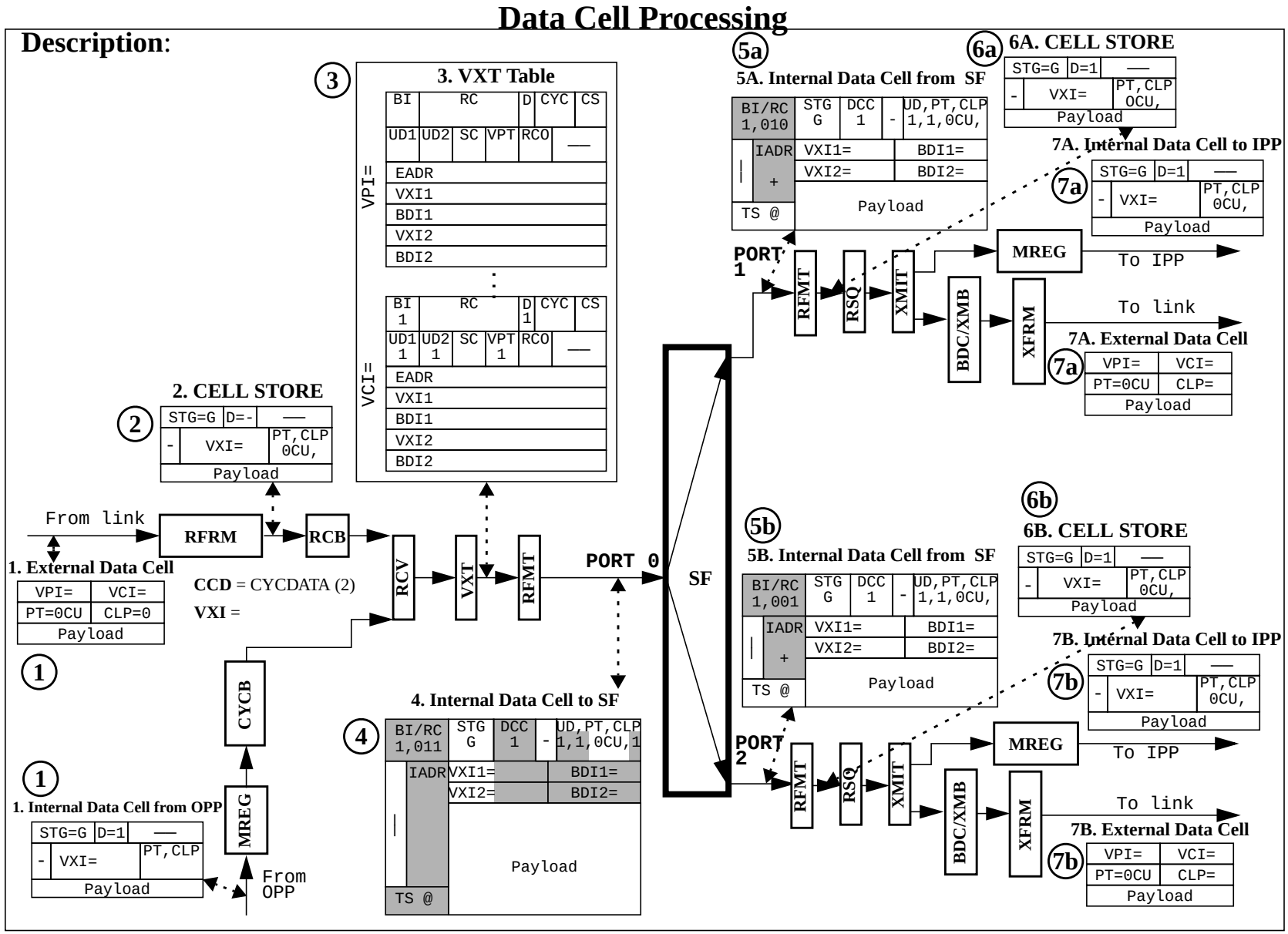


Figure 38: Data Cell Scenario Template (VC, not VP)

Software Control of WUGS Switch Chips

J. Andrew Fingerhut

Washington University
Saint Louis, Missouri

WUGS Switch Operational Scenarios

J. Andrew Fingerhut

Washington University
Saint Louis, Missouri