Primary Vertex dZ Cut Firmware for CMS p2gt

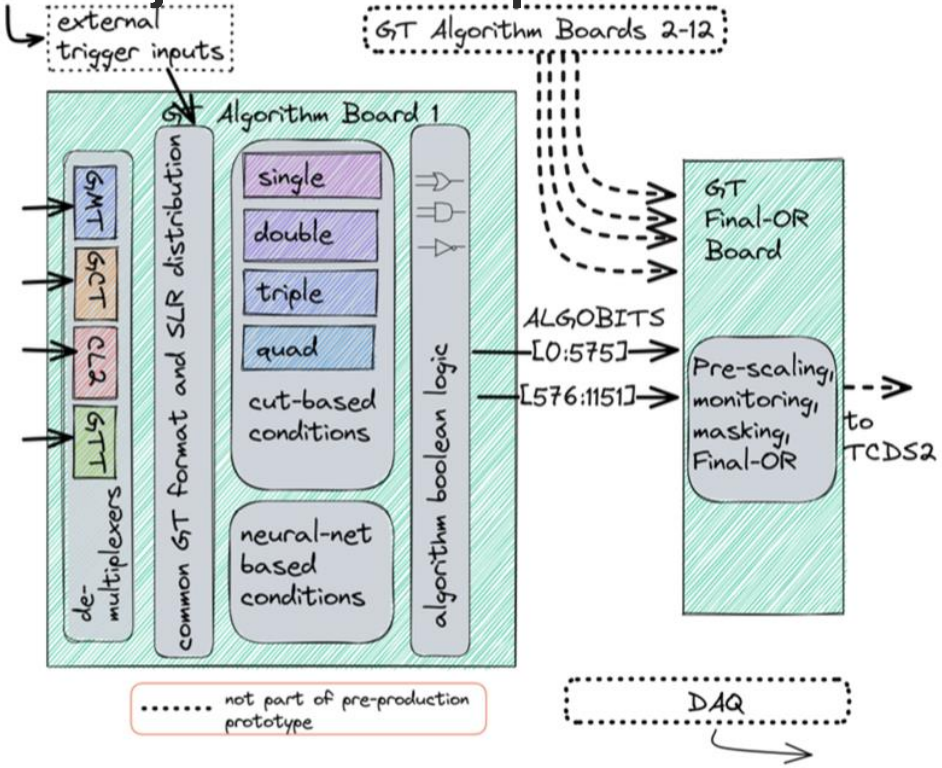
Claire Lundy

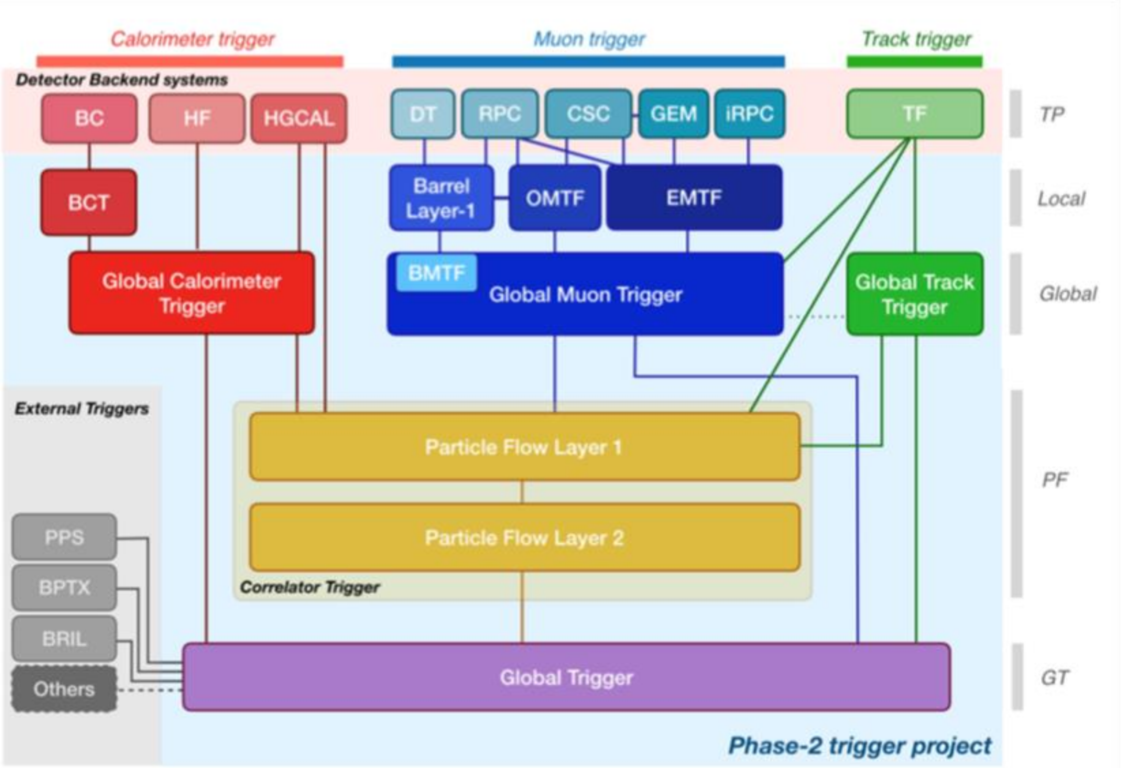
August 10th, 2023

Background

- luminosity upgrade => CMS L1T upgrade!

- GT applies cut-based algorithms to incoming object data and passes results to HLT





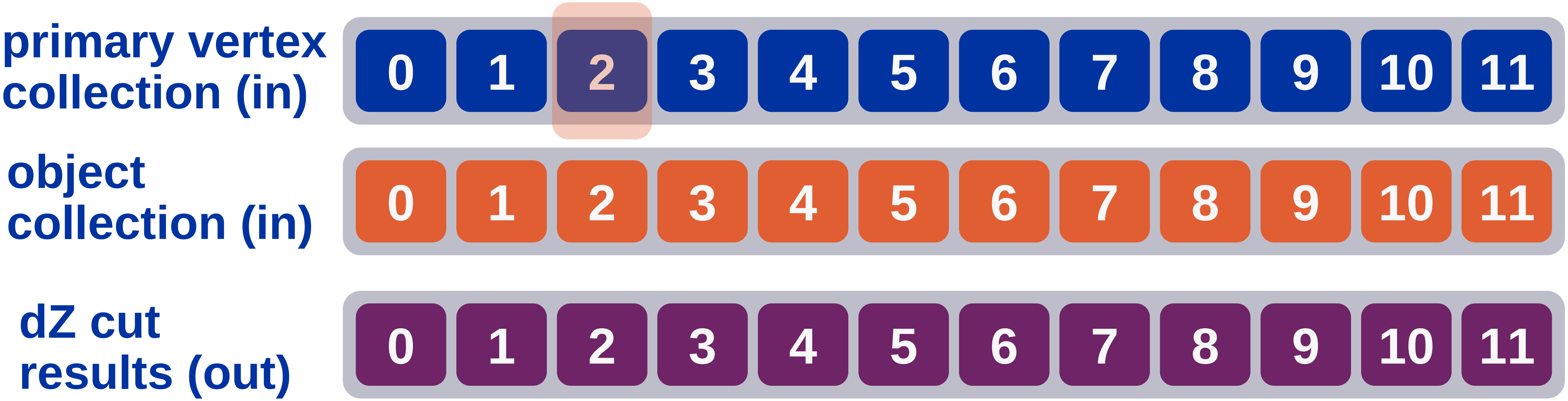
- within these cuts, we want one that finds the absolute difference in z-position between an object and a primary vertex

Primary Vertex dZ Cut

1. Grab pv z0 position from collection based on an index (0-11).

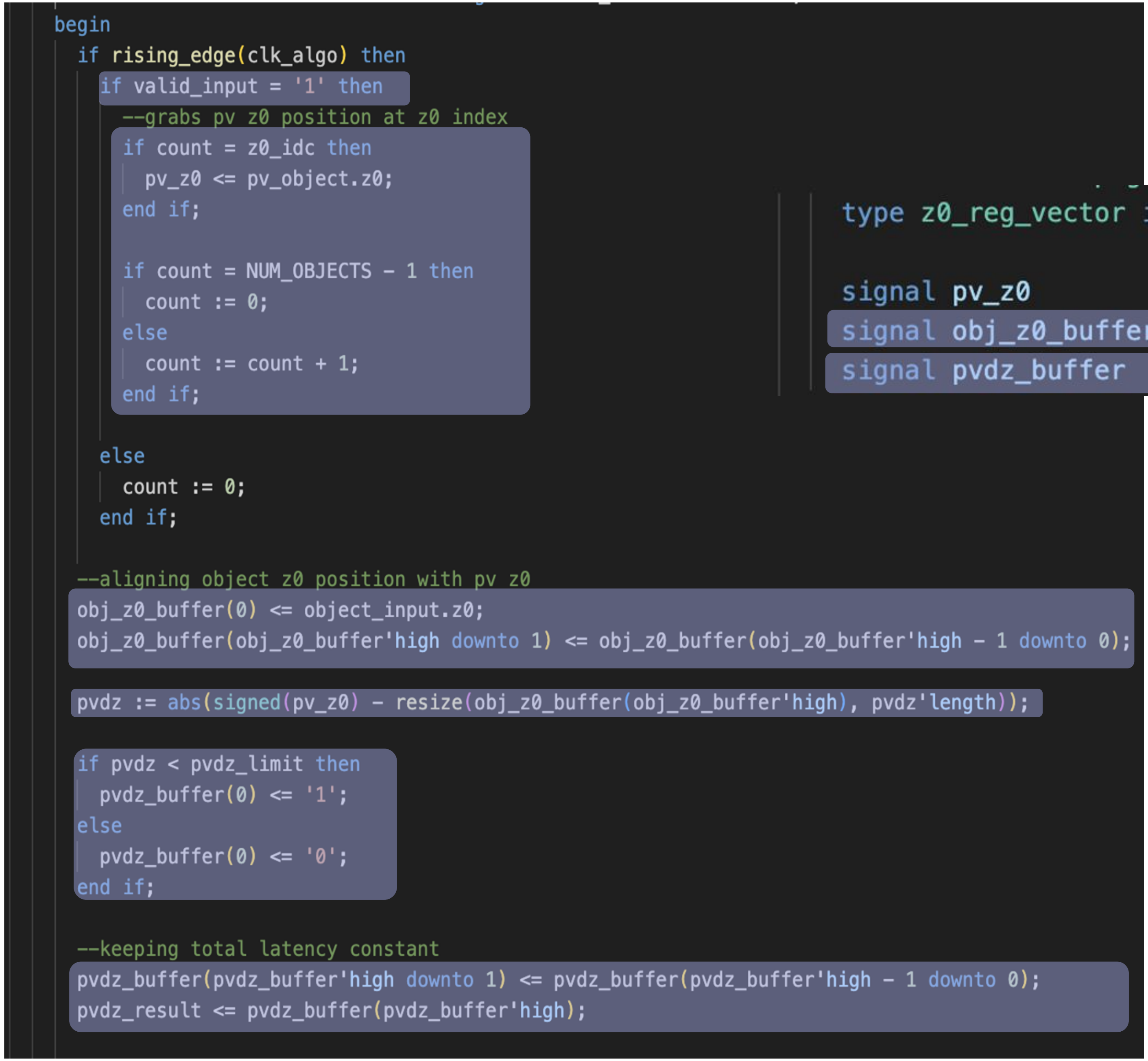
2. Compute |pv.z0 - obj.z0| for entire 12 obj collection.

3. Compare abs difference to a limit. Send 1 if it’s within the limit, else 0.

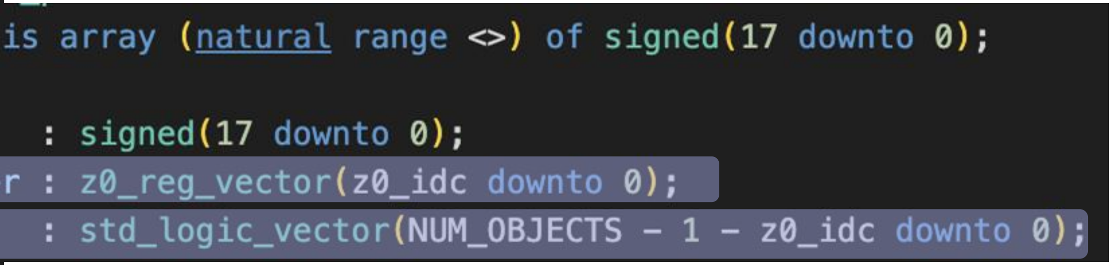


primary vertex collection (in)	0	1	2	3	4	5	6	7	8	9	10	11
object collection (in)	0	1	2	3	4	5	6	7	8	9	10	11
dZ cut results (out)	0	1	2	3	4	5	6	7	8	9	10	11

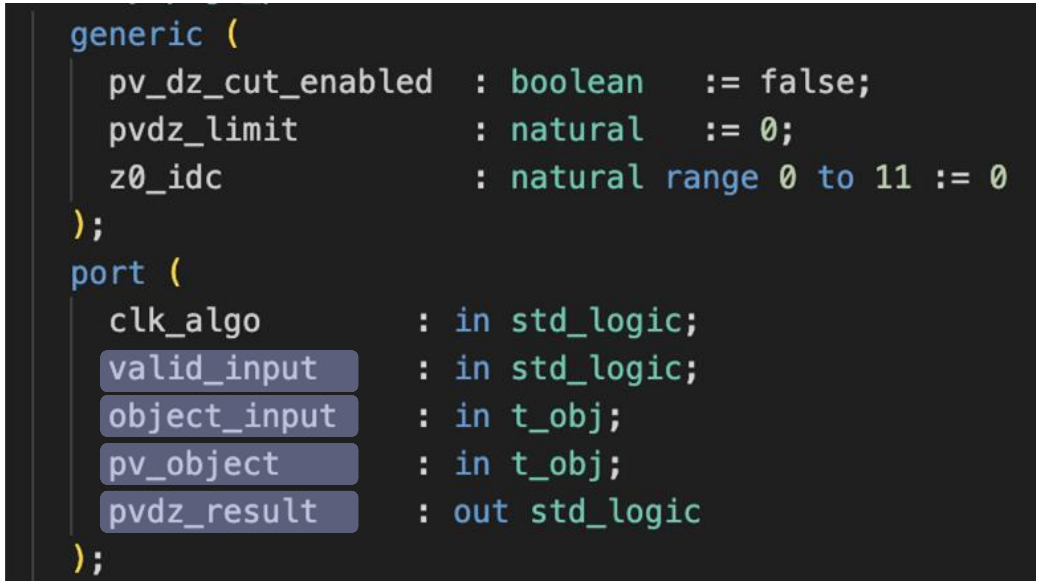
dZ Cut Module



```
begin
  if rising_edge(clk_algo) then
    if valid_input = '1' then
      --grabs pv z0 position at z0 index
      if count = z0_idc then
        pv_z0 <= pv_object.z0;
      end if;

      if count = NUM_OBJECTS - 1 then
        count := 0;
      else
        count := count + 1;
      end if;

    else
      count := 0;
    end if;

    --aligning object z0 position with pv z0
    obj_z0_buffer(0) <= object_input.z0;
    obj_z0_buffer(obj_z0_buffer'high downto 1) <= obj_z0_buffer(obj_z0_buffer'high - 1 downto 0);

    pvdz := abs(signed(pv_z0) - resize(obj_z0_buffer(obj_z0_buffer'high), pvdz'length));

    if pvdz < pvdz_limit then
      pvdz_buffer(0) <= '1';
    else
      pvdz_buffer(0) <= '0';
    end if;

    --keeping total latency constant
    pvdz_buffer(pvdz_buffer'high downto 1) <= pvdz_buffer(pvdz_buffer'high - 1 downto 0);
    pvdz_result <= pvdz_buffer(pvdz_buffer'high);
```

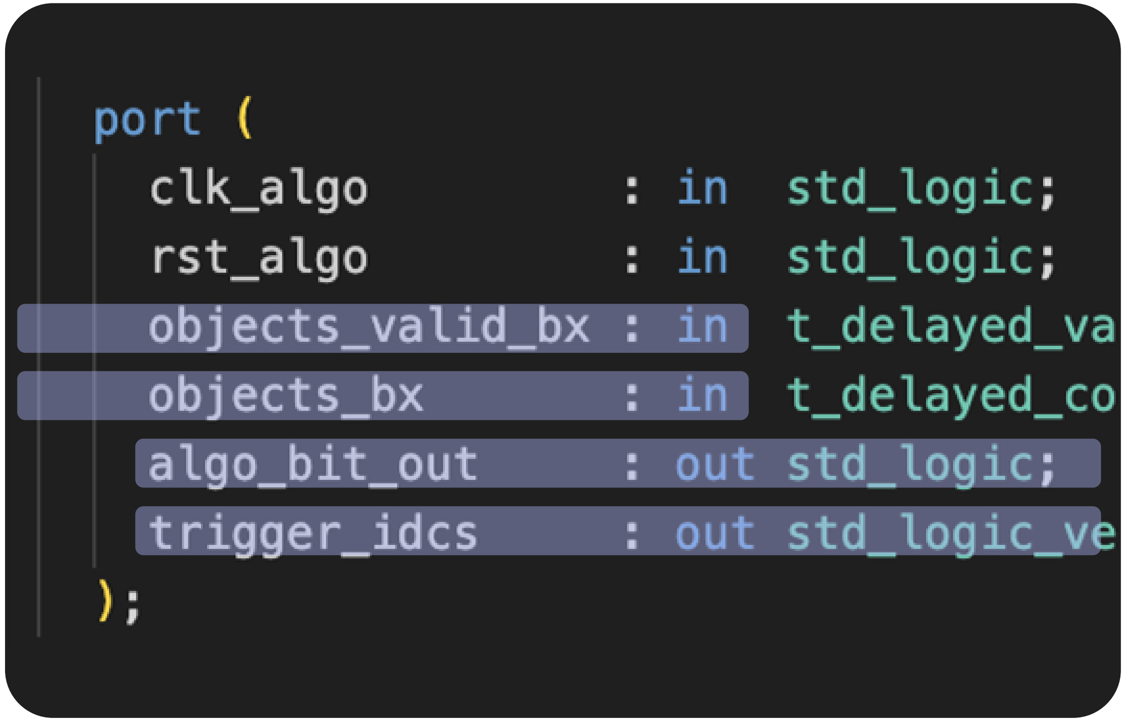


```
type z0_reg_vector is array (natural range <>) of signed(17 downto 0);

signal pv_z0          : signed(17 downto 0);
signal obj_z0_buffer : z0_reg_vector(z0_idc downto 0);
signal pvdz_buffer    : std_logic_vector(NUM_OBJECTS - 1 - z0_idc downto 0);
```

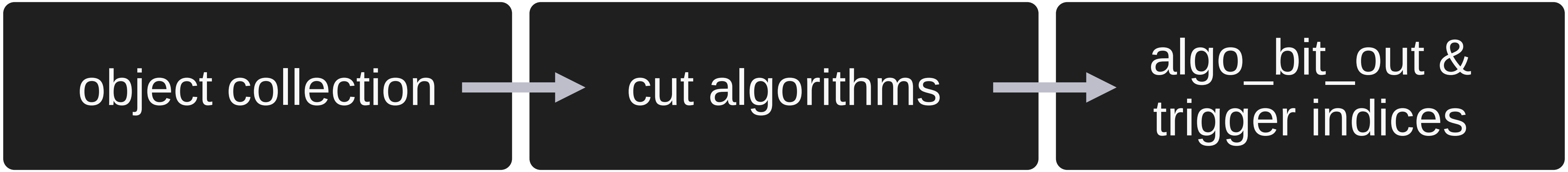


```
generic (
  pv_dz_cut_enabled  : boolean   := false;
  pvdz_limit         : natural   := 0;
  z0_idc             : natural range 0 to 11 := 0
);
port (
  clk_algo         : in std_logic;
  valid_input      : in std_logic;
  object_input     : in t_obj;
  pv_object        : in t_obj;
  pvdz_result      : out std_logic
);
```

singleObjCondition

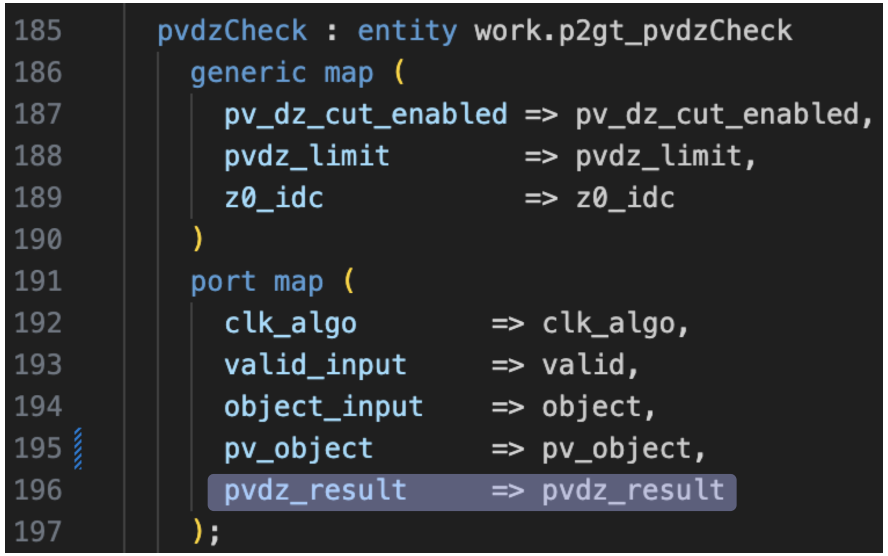


```
port (
    clk_algo          : in   std_logic;
    rst_algo          : in   std_logic;
    objects_valid_bx  : in   t_delayed_va
    objects_bx        : in   t_delayed_co
    algo_bit_out      : out  std_logic;
    trigger_idcs      : out  std_logic_ve
);
```

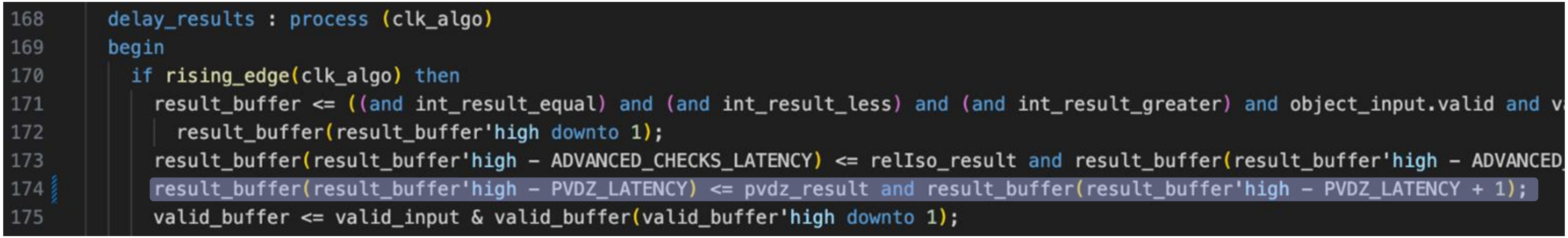


object collection → cut algorithms → algo_bit_out & trigger indices

singleObjCond Integration

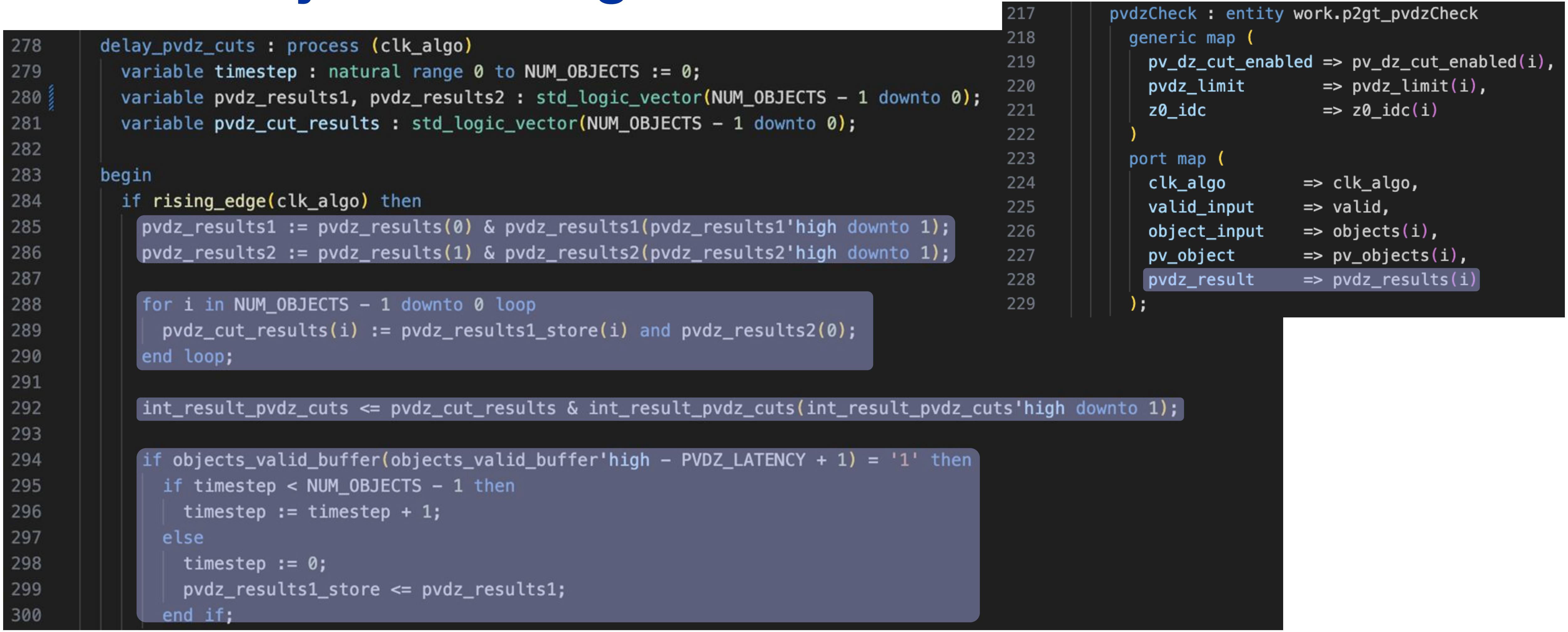


```
pvdzCheck : entity work.p2gt_pvdzCheck
  generic map (
    pv_dz_cut_enabled => pv_dz_cut_enabled,
    pvdz_limit        => pvdz_limit,
    z0_idc            => z0_idc
  )
  port map (
    clk_algo        => clk_algo,
    valid_input     => valid,
    object_input    => object,
    pv_object       => pv_object,
    pvdz_result     => pvdz_result
  );
```



```
delay_results : process (clk_algo)
begin
  if rising_edge(clk_algo) then
    result_buffer <= ((and int_result_equal) and (and int_result_less) and (and int_result_greater) and object_input.valid and v
      result_buffer(result_buffer'high downto 1);
    result_buffer(result_buffer'high - ADVANCED_CHECKS_LATENCY) <= relIso_result and result_buffer(result_buffer'high - ADVANCED
    result_buffer(result_buffer'high - PVDZ_LATENCY) <= pvdz_result and result_buffer(result_buffer'high - PVDZ_LATENCY + 1);
    valid_buffer <= valid_input & valid_buffer(valid_buffer'high downto 1);
```

doubleObjCond Integration



```
delay_pvdz_cuts : process (clk_algo)
  variable timestep : natural range 0 to NUM_OBJECTS := 0;
  variable pvdz_results1, pvdz_results2 : std_logic_vector(NUM_OBJECTS - 1 downto 0);
  variable pvdz_cut_results : std_logic_vector(NUM_OBJECTS - 1 downto 0);

begin
  if rising_edge(clk_algo) then
    pvdz_results1 := pvdz_results(0) & pvdz_results1(pvdz_results1'high downto 1);
    pvdz_results2 := pvdz_results(1) & pvdz_results2(pvdz_results2'high downto 1);

    for i in NUM_OBJECTS - 1 downto 0 loop
      pvdz_cut_results(i) := pvdz_results1_store(i) and pvdz_results2(0);
    end loop;

    int_result_pvdz_cuts <= pvdz_cut_results & int_result_pvdz_cuts(int_result_pvdz_cuts'high downto 1);

    if objects_valid_buffer(objects_valid_buffer'high - PVDZ_LATENCY + 1) = '1' then
      if timestep < NUM_OBJECTS - 1 then
        timestep := timestep + 1;
      else
        timestep := 0;
        pvdz_results1_store <= pvdz_results1;
      end if;
```

```
pvdzCheck : entity work.p2gt_pvdzCheck
  generic map (
    pv_dz_cut_enabled => pv_dz_cut_enabled(i),
    pvdz_limit        => pvdz_limit(i),
    z0_idc            => z0_idc(i)
  )
  port map (
    clk_algo        => clk_algo,
    valid_input     => valid,
    object_input    => objects(i),
    pv_object       => pv_objects(i),
    pvdz_result     => pvdz_results(i)
  );
```

That’s all!

- dZ module is now written, integrated into single, double, and triple object condition files, and tested for each
- Missing for the quadruple object condition, but the framework is there…just needs a bit more testing.
- Big thanks to my supervisor, Benjamin Huber, and the P2GT group–it was a really cool project and a fun summer :)

Thanks!

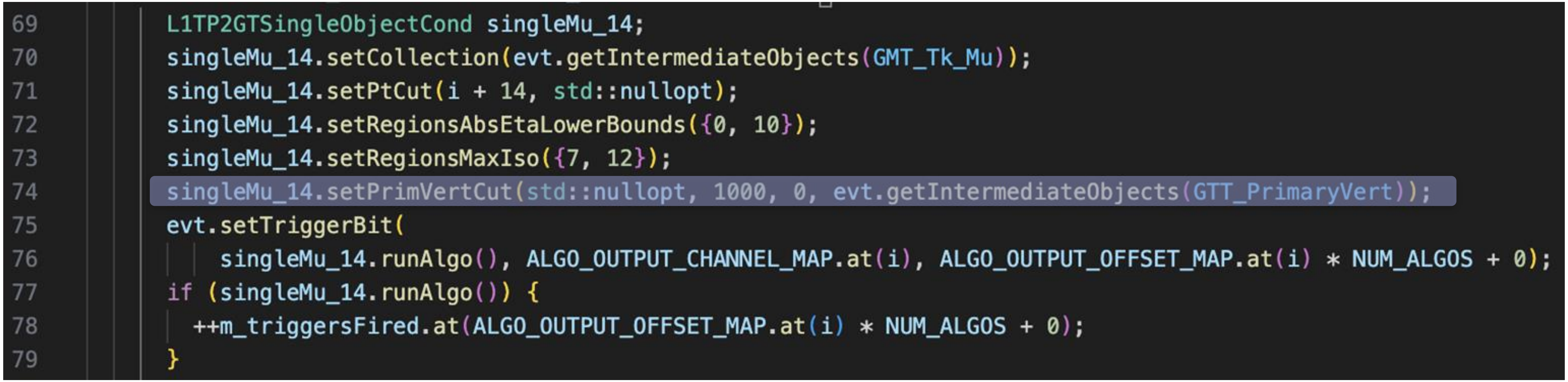
Karl-Marx-Str.
186 198
Jonasstr.
NO SELFIE STICKS OVER THE FENCE

References

Bortolato, G., Cepeda, M., Heikkilä, J., Huber, B., Leutgeb, E., Rabady, D., & Sakulin, H. (2023). *The Level-1 Global Trigger for Phase-2: Algorithms, configuration and integration in the CMS offline framework*.

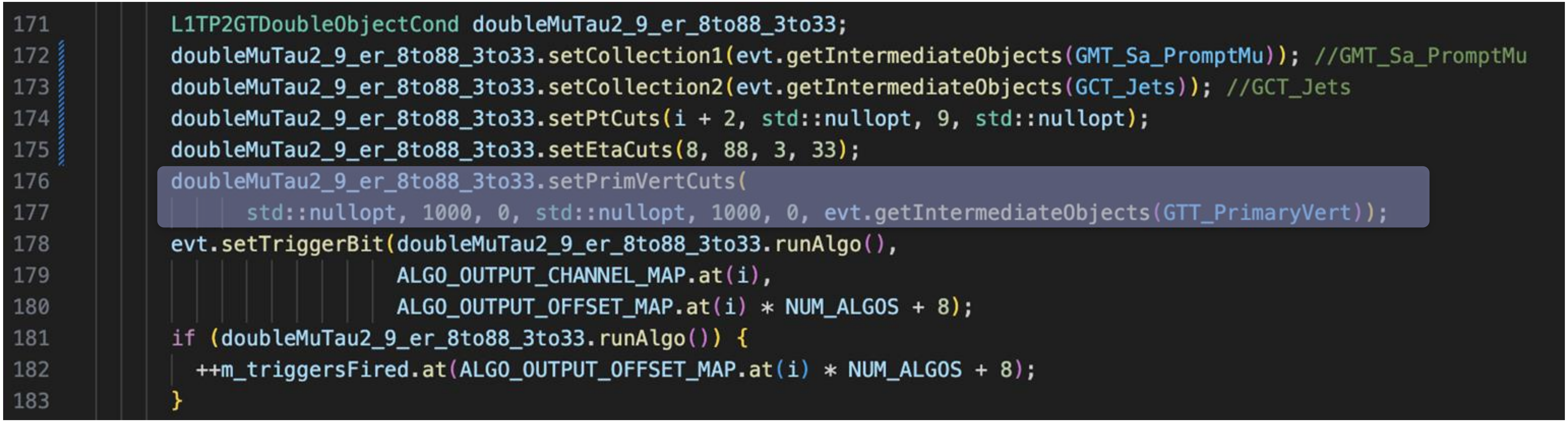
CMS Collaboration. (2020). *The Phase-2 Upgrade of the CMS Level-1 Trigger*. 121–129, 188–194.

singleObjCond testing



```
L1TP2GTSingleObjectCond singleMu_14;
singleMu_14.setCollection(evt.getIntermediateObjects(GMT_Tk_Mu));
singleMu_14.setPtCut(i + 14, std::nullopt);
singleMu_14.setRegionsAbsEtaLowerBounds({0, 10});
singleMu_14.setRegionsMaxIso({7, 12});
singleMu_14.setPrimVertCut(std::nullopt, 1000, 0, evt.getIntermediateObjects(GTT_PrimaryVert));
evt.setTriggerBit(
    singleMu_14.runAlgo(), ALGO_OUTPUT_CHANNEL_MAP.at(i), ALGO_OUTPUT_OFFSET_MAP.at(i) * NUM_ALGOS + 0);
if (singleMu_14.runAlgo()) {
  ++m_triggersFired.at(ALGO_OUTPUT_OFFSET_MAP.at(i) * NUM_ALGOS + 0);
}
```

doubleObjCond testing



```
L1TP2GTDoubleObjectCond doubleMuTau2_9_er_8to88_3to33;
doubleMuTau2_9_er_8to88_3to33.setCollection1(evt.getIntermediateObjects(GMT_Sa_PromptMu)); //GMT_Sa_PromptMu
doubleMuTau2_9_er_8to88_3to33.setCollection2(evt.getIntermediateObjects(GCT_Jets)); //GCT_Jets
doubleMuTau2_9_er_8to88_3to33.setPtCuts(i + 2, std::nullopt, 9, std::nullopt);
doubleMuTau2_9_er_8to88_3to33.setEtaCuts(8, 88, 3, 33);
doubleMuTau2_9_er_8to88_3to33.setPrimVertCuts(
      std::nullopt, 1000, 0, std::nullopt, 1000, 0, evt.getIntermediateObjects(GTT_PrimaryVert));
evt.setTriggerBit(doubleMuTau2_9_er_8to88_3to33.runAlgo(),
                  ALGO_OUTPUT_CHANNEL_MAP.at(i),
                  ALGO_OUTPUT_OFFSET_MAP.at(i) * NUM_ALGOS + 8);
if (doubleMuTau2_9_er_8to88_3to33.runAlgo()) {
  ++m_triggersFired.at(ALGO_OUTPUT_OFFSET_MAP.at(i) * NUM_ALGOS + 8);
}
```