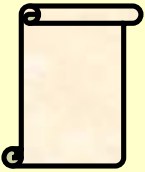


DØ Level 2 Global Processor



**R. Moore,
Michigan State University**

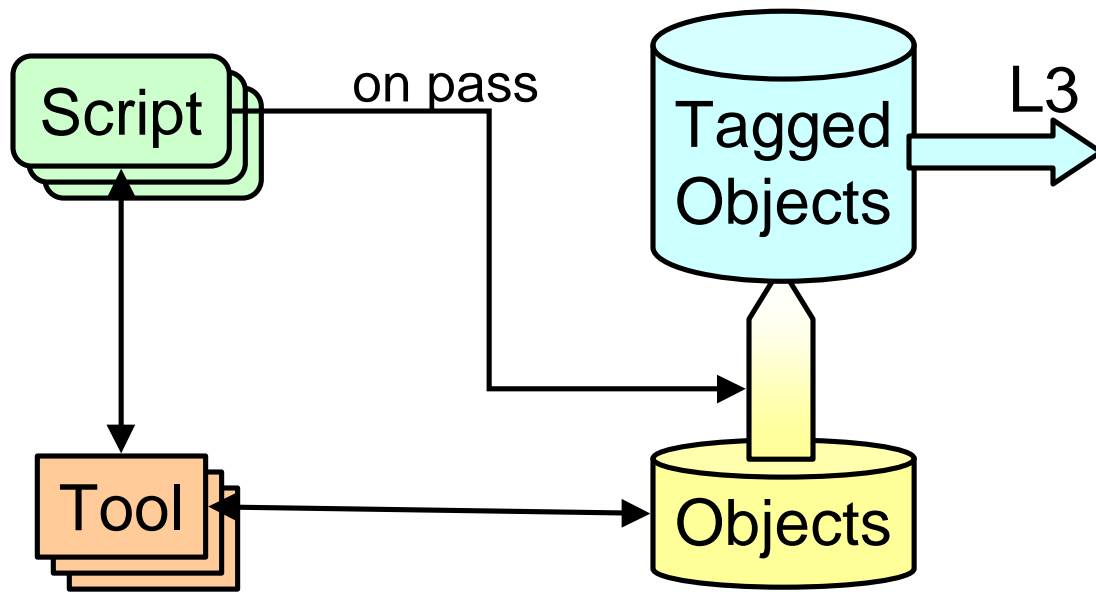


Basic Design



- Global worker

- verifies each of (128 max) L1 bits which are set
- has one script per bit
- scripts call tools which find physics objects
- objects used to pass event tagged and saved

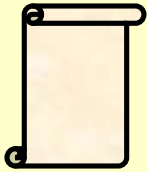


Initialization

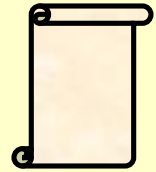
- **Common definition file for L1, L2 and L3 triggers**
 - **parsed and checked by COOR**
 - **outputs low level, L2-only text file to TCC**
 - **TCC converts this to alpha binary for download to Bit3**
 - **Alphas only need version check on configuration info**
 - **avoids complex parser in alpha environment**

Initialization II

- **Features of definition file**
 - **define tools and then use them to build scripts (just like code)**
 - **fakes multiple L2 scripts per L1 trigger bit by duplicating L1 bits behind the scenes**
 - **includes checks to ensure L2 and L3 readout lists are compatible (MBT fixed readout pattern)**



The Script



- **Scripts contain a list of tool pointers with parameters to call them with**
 - **scripts know nothing about the tools they call**
 - **tools return a token which is used to tag their objects if the script passes**
- **Default behaviour of empty scripts is to pass events**
- **Each script stores pass/fail statistics for each tool**
 - **output on ‘collect status’**



The Tool



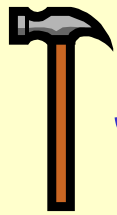
- **Tools create physics objects from the input data**
 - **data is ordered to reduce search times**
 - **objects kept for reuse on multiple calls**
 - **some parameters defined at initialization others when tool called**



Timing Studies



- **Time to process event in Global worker involves**
 - searching for the trigger bits
 - **calling overhead (script® tool)**
 - reading and writing data (I/O)
 - **running algorithms**
 - servicing “new event” interrupt (same for all L2 workers)
- **Compared speeds of different programming languages**

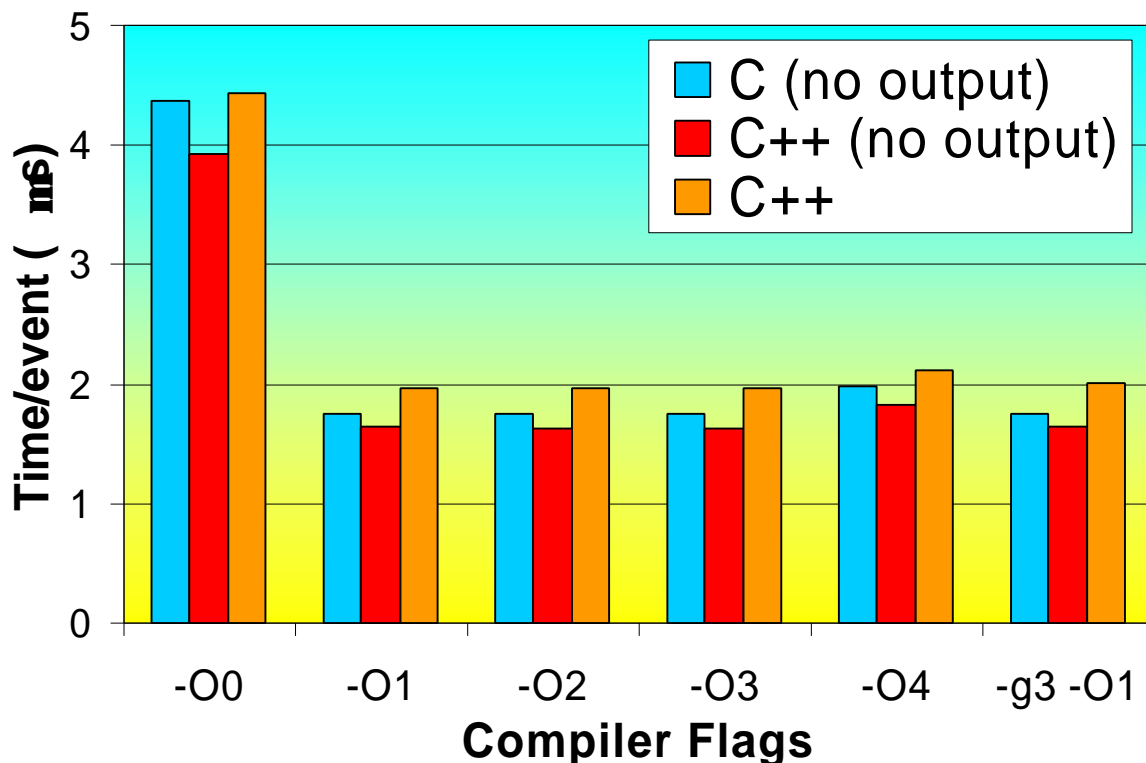


Single Tool Timing

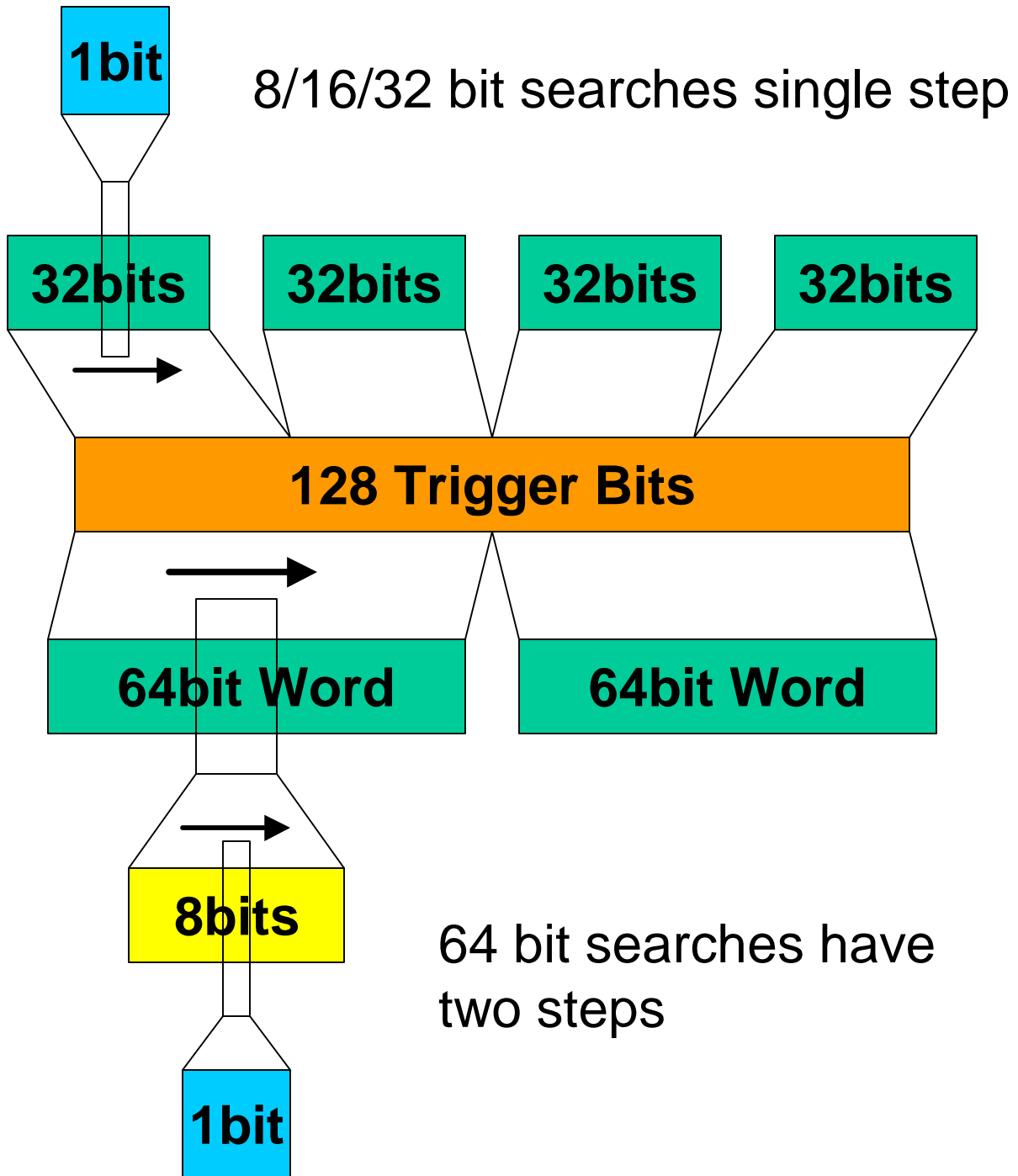


- Tested relative times of an electron tool (433MHz CPU)
 - one tool in one script
 - used electrons in Run I events
 - 1000 event sample size
 - 22.9% events written out (pass)

⇒ 0.15ms/tool output time @ 10%

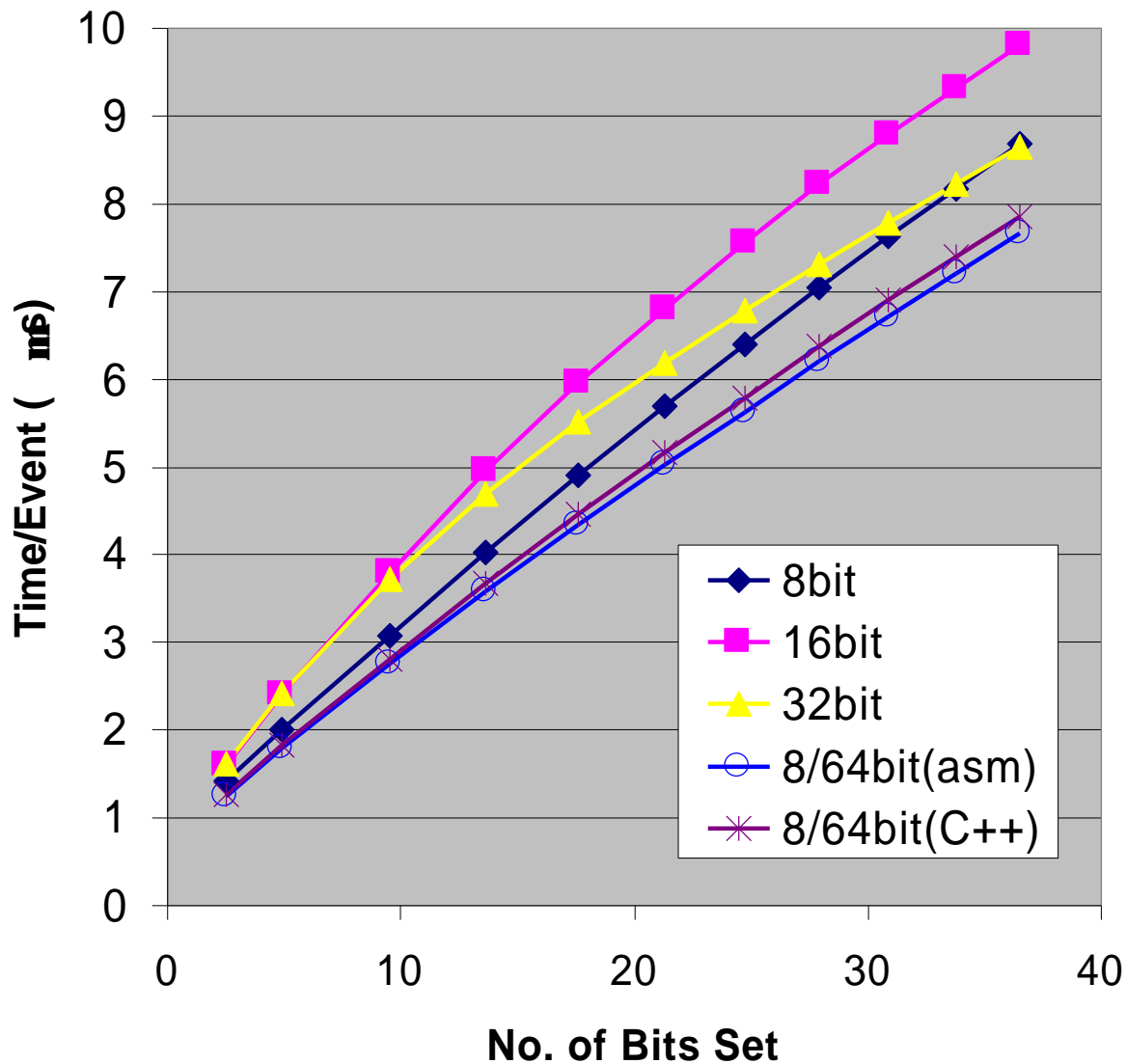


Searching for Bits



Bit Search Times

- Times include calling single script containing one empty tool



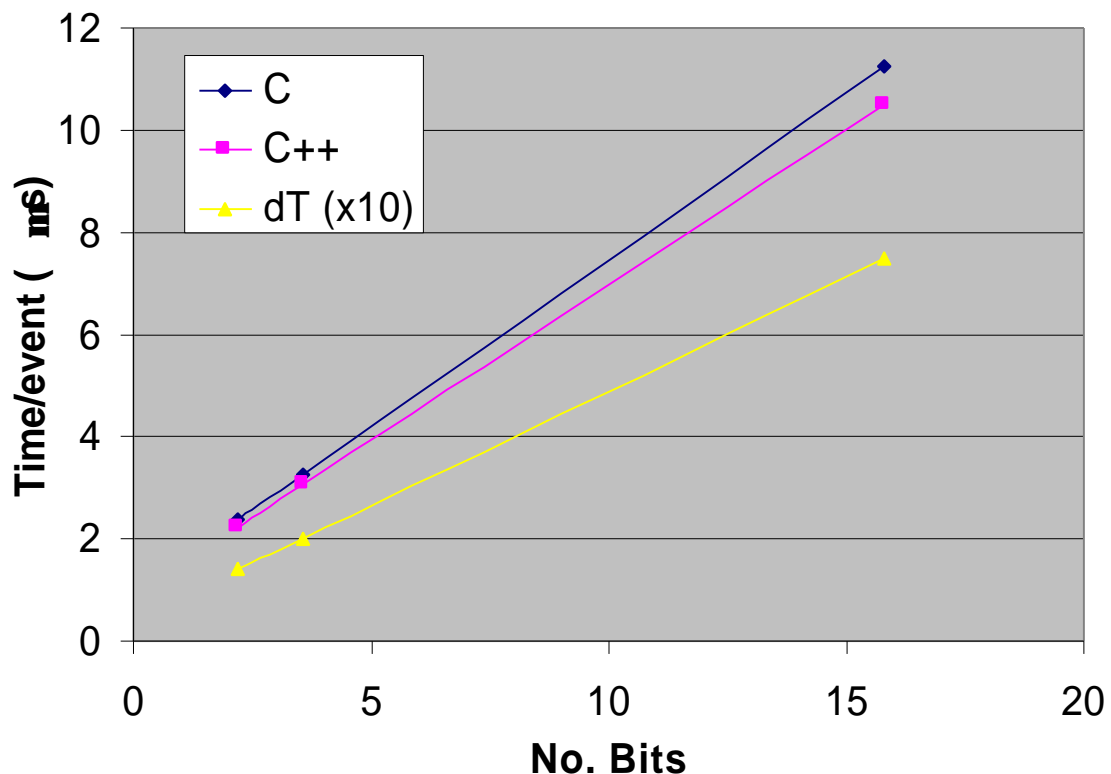
Framework Timing

- **C and C++ style L2 Script Runners written and timed**
 - 128 scripts
 - 50 “pass” tools with fixed pass probability
 - tools/script taken from Run I trigger configuration
 - L1 trigger bits from Run I data used (biased because events passed L3)
- **Pass tool threshold adjusted so that ~10% events passed**

Framework Timing II

- **Features**

- **points show estimated Run I and Run II bits/event and Run I “L3 passed” bits/event**
- **times include random number generation**



Timing Summary

- **Summary assumes 5 bits average per event**
 - double counts call overhead
 - includes random number generation in framework test
- **Should be compared to Run I**
 - ~1.5 tools/script

Timing Budget	50ms
Framework + bit search	-4 μ s
Interrupt (estimate)	-2 μ s
Remaining budget	44ms
Time/tool (inc. I/O)	2 μ s
Max Tools callable	22
Max Tools/script	4.4

Conclusions/Status

- Chosen C++ over C
 - no dynamic memory allocation
 - no RTTI
 - no C++-style exceptions
 - little/no STL (due to above)
- Prototype framework written
 - several design changes to add
 - e.g. improve I/O for “big” events
- Next...
 - add global to trigger simulation
 - integrate prototype tools into framework and test using simulation data