connect ONNX to every deep learning accelerator

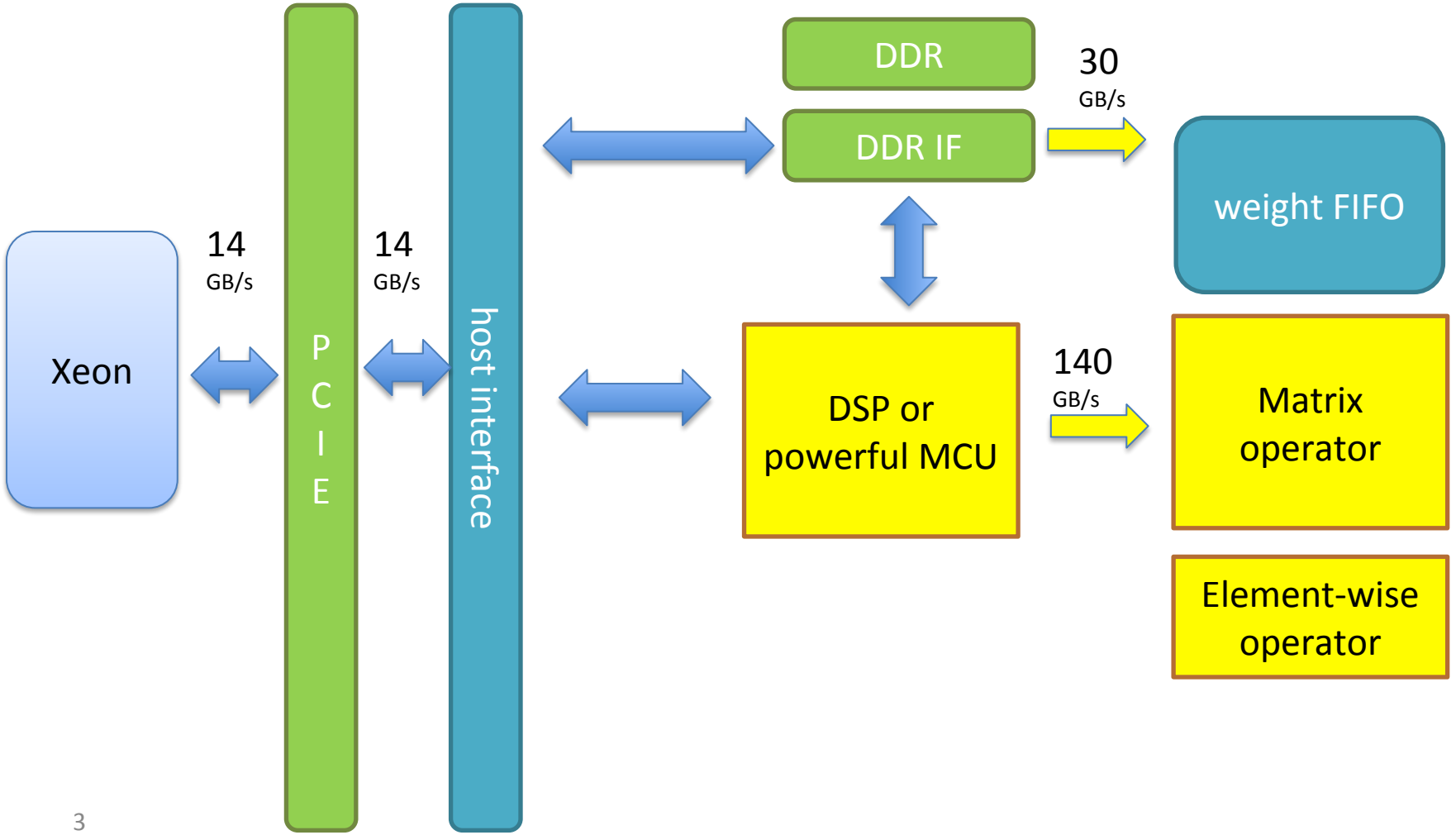
https://onnc.ai

ONNC

traditional compiler

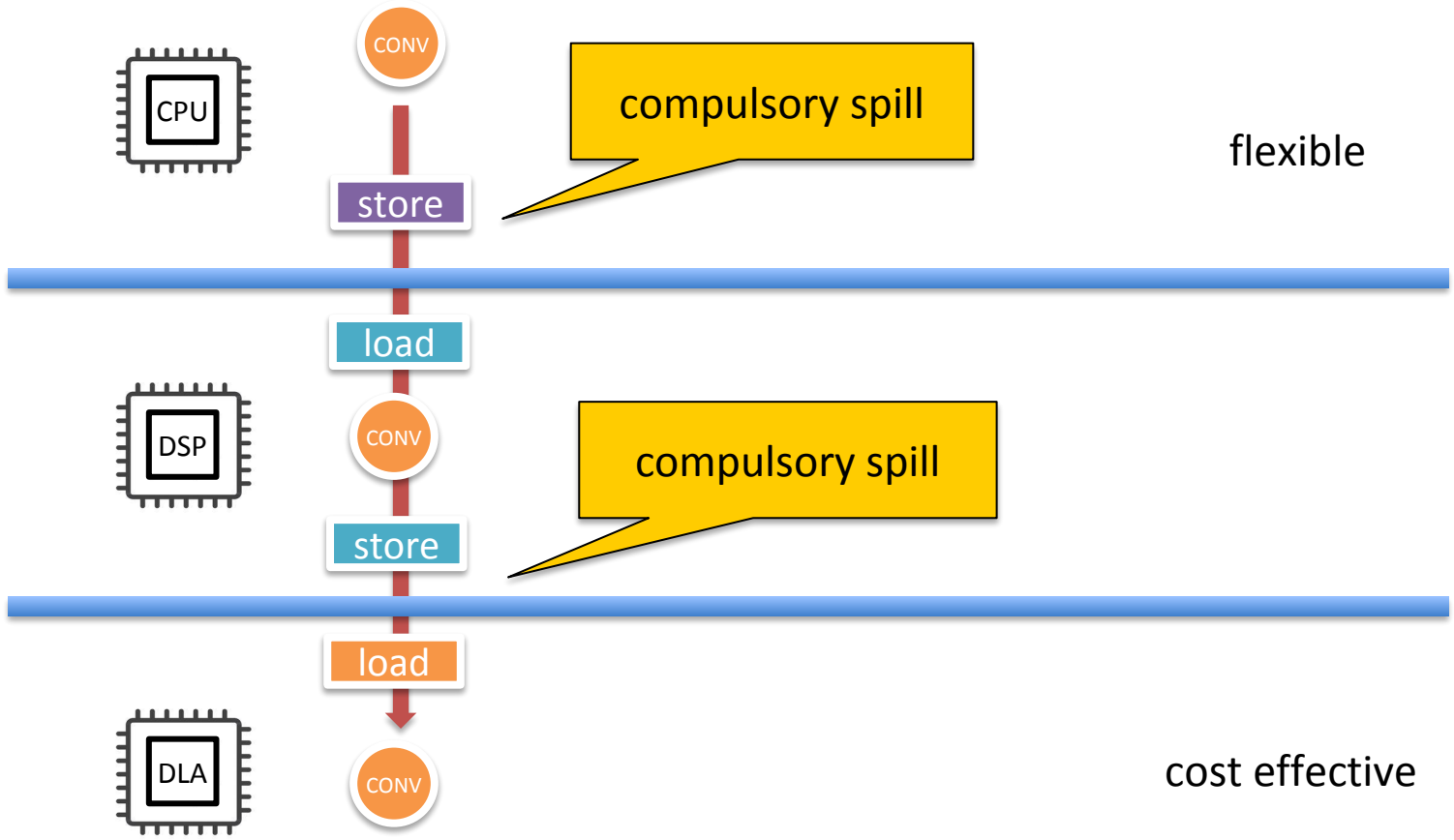
target	heterogeneous architecture	single architecture
programming model	PetriNet(1)	CFG and DFG
IR type	ONNX IR (multiple outputs)	three address code (single output)
physical feature depends on	operand	opcode

Heterogeneous System to well control the data flow



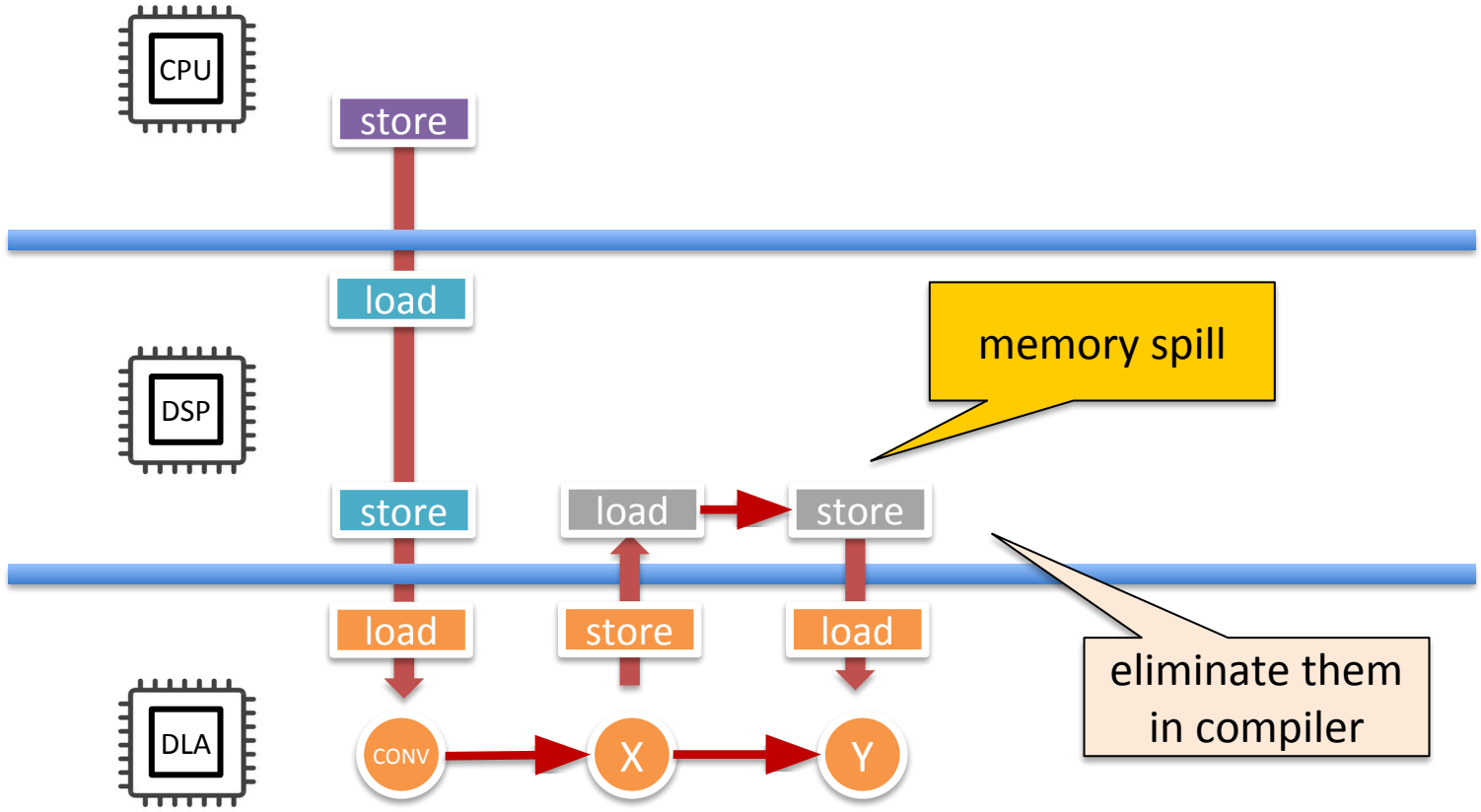
Compulsory Spill

is easy to implement in the other compiler framework



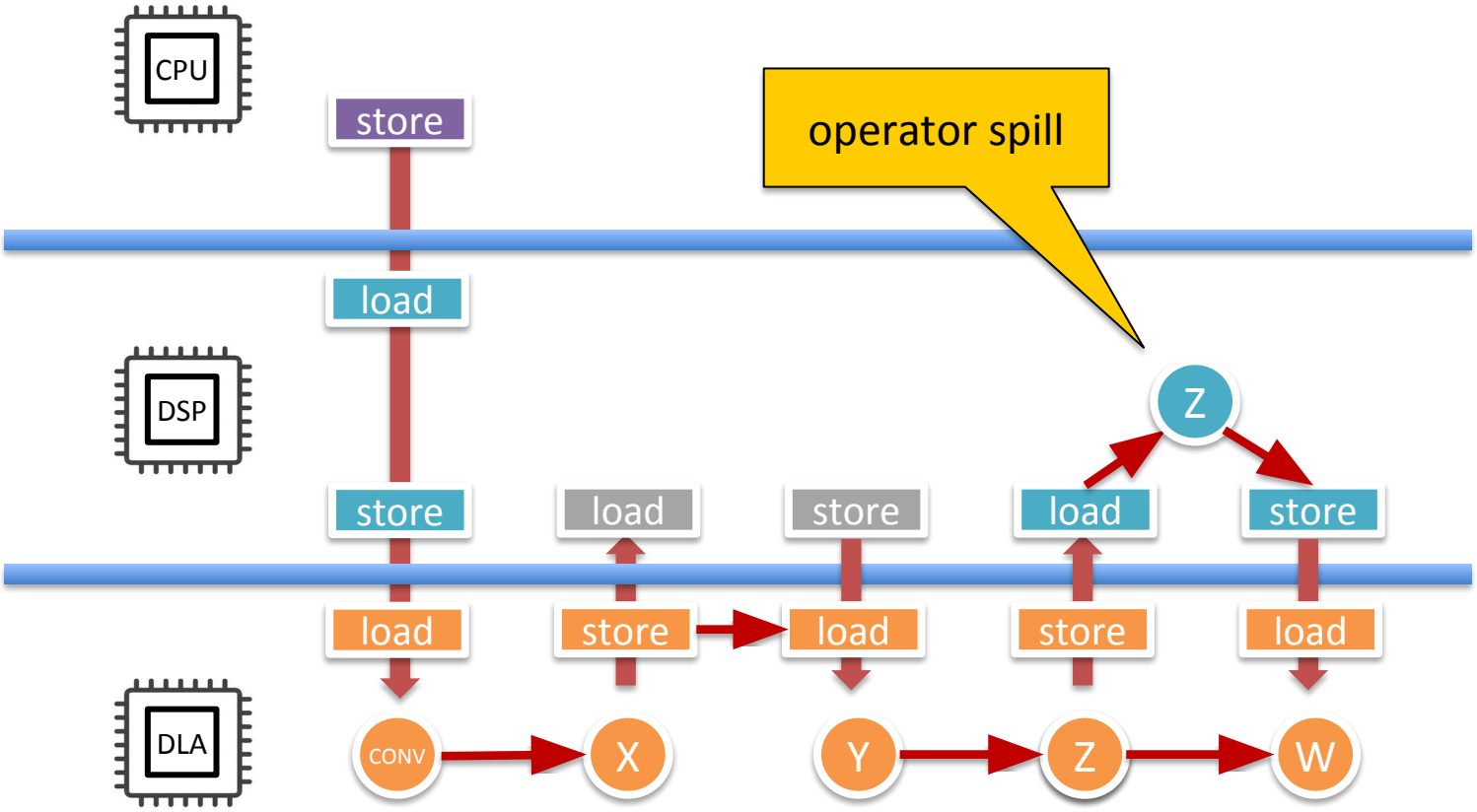
Memory Spill

is what we already have in every compiler framework



Operator Spill

is totally new and required for every accelerators



	ONNC	 traditional compiler
target	heterogeneous architecture system (HSA)	single architecture
programming model	PetriNet(1)	CFG and DFG
IR type	ONNX IR (multiple outputs)	three address code (single output)
physical feature depends on	operand	opcode

ONNC

traditional compiler

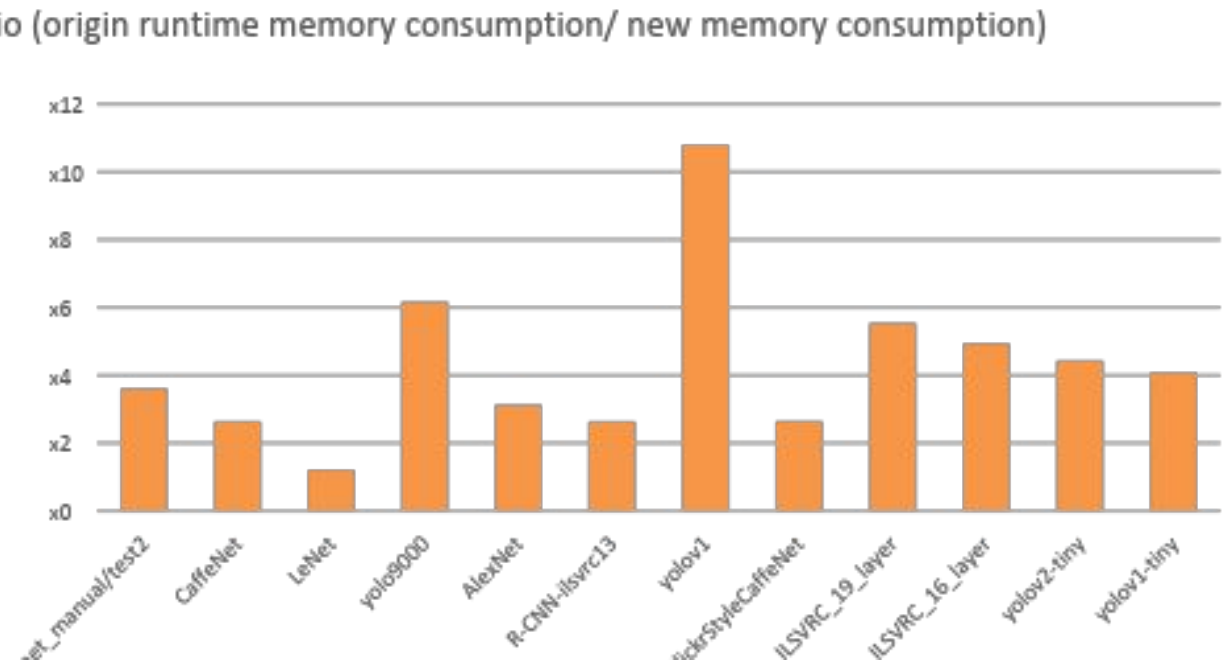
Memory runtime consumption

Limited DLA
save x3.77 in avg.

paging system

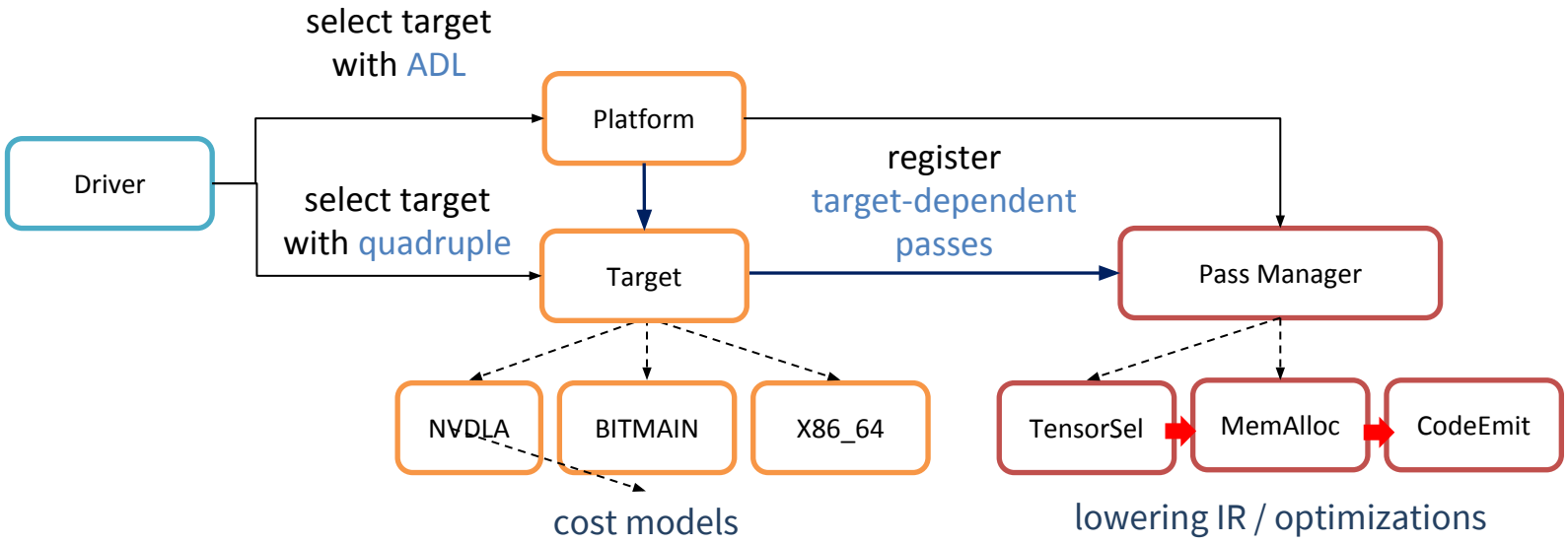
- ONNC reuses local memory and save runtime memory consumption by life range analysis of tensors
- The runtime memory consumption includes inputs, outputs and weights

Experimental environment
Ubuntu Linux 16.04
Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz
32G RAM



ONNC supports various target devices

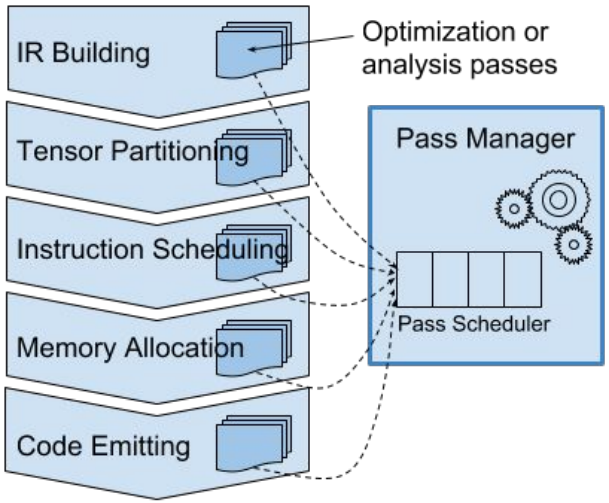
- Use LLVM-like triple to select compiler target backend
  - compiler
  - loader
  - calibration
- Use architecture description to select target platform
  - A platform contains multiple target backends
- Platform and Target instance register target-dependent passes into PassManager



skymizer

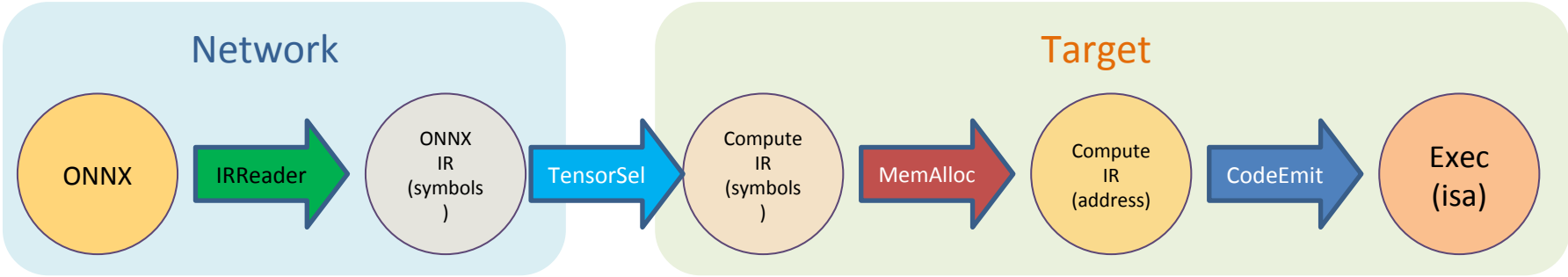
Others

process control | Pass manager | None. Simple linked list



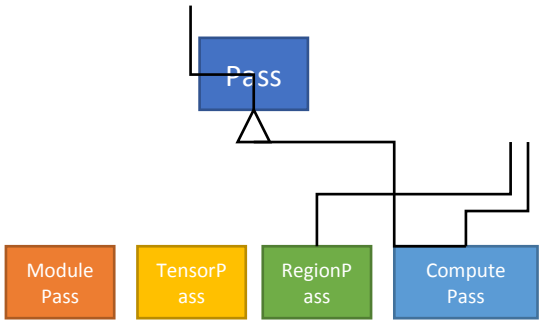
Transformation: from Compiler's point of view

- Target phases
  - TensorSel - turn ONNX layer to target-specific operator
  - MemAlloc - turn symbols of tensor to memory address
  - CodeEmit - turn IR to machine instruction
- Platform phases
  - TensorPartition - map subgraph to hardware processing unit
  - OprSched - find the timing sequence of a subgraph



Four Kinds of Passes in ONNC

- ModulePass
  - The most general of all superclasses that you can use
  - Use entire network as an unit
- TensorPass
  - Use Tensor Graph as an unit
  - Tensor Graph bases on ONNX IR
- RegionPass
  - Use each signle-entry-signe-exit region in a tensor graph as an unit
  - For example, groups in GoogLeNet
- ComputePass
  - Use Compute Graph as an unit




```
// methods in class Pass
bool run(Module& pModule);
virtual bool doInitialization(Module& pModule);
virtual bool doFinalization(Module& pModule);
```

```
// methods in class ModulePass
virtual bool runOnModule(Module& pModule) = 0;

// methods in class TensorPass
virtual bool runOnTensor(TensorGraph& pGraph) = 0;
```

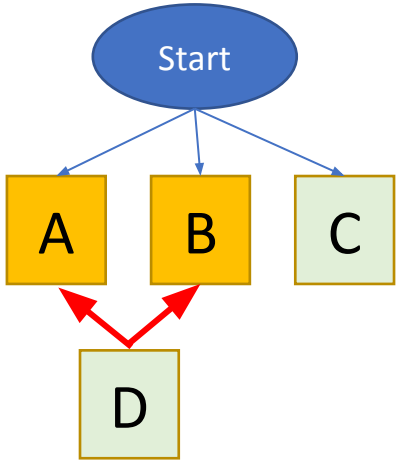
AnalysisUsage describes the dependencies between Passes

- PassManager automatically creates all Passes that used by the other Passes.
- Similar to LLVM. Engineers who already familiar to LLVM can understand ONNC in a short time

```
/// in A.cpp
INITIALIZE_PASS(A, “pass_a”)
```

```
/// in B.cpp
INITIALIZE_PASS(B, “pass_b”)
```

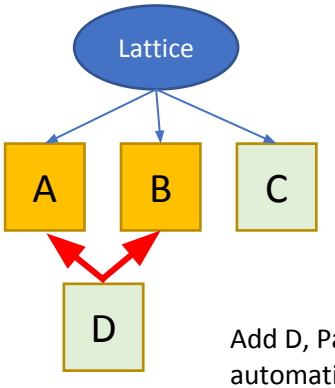
```
/// methods in Pass D. Override
void D::getAnalysisUsage(AnalysisUsage& pUsage) const
{
  pUsage.addRequiredID(A::ID);
  PUsage.addRequiredID(B::ID);
}
```



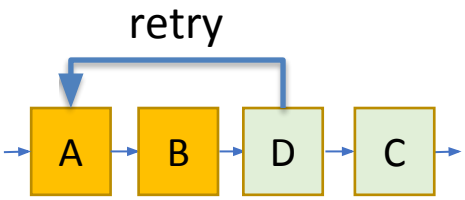
Innate Iterative Compilation

traditional compiler

compilation model	ITERATIVE	sequential



topologic sort

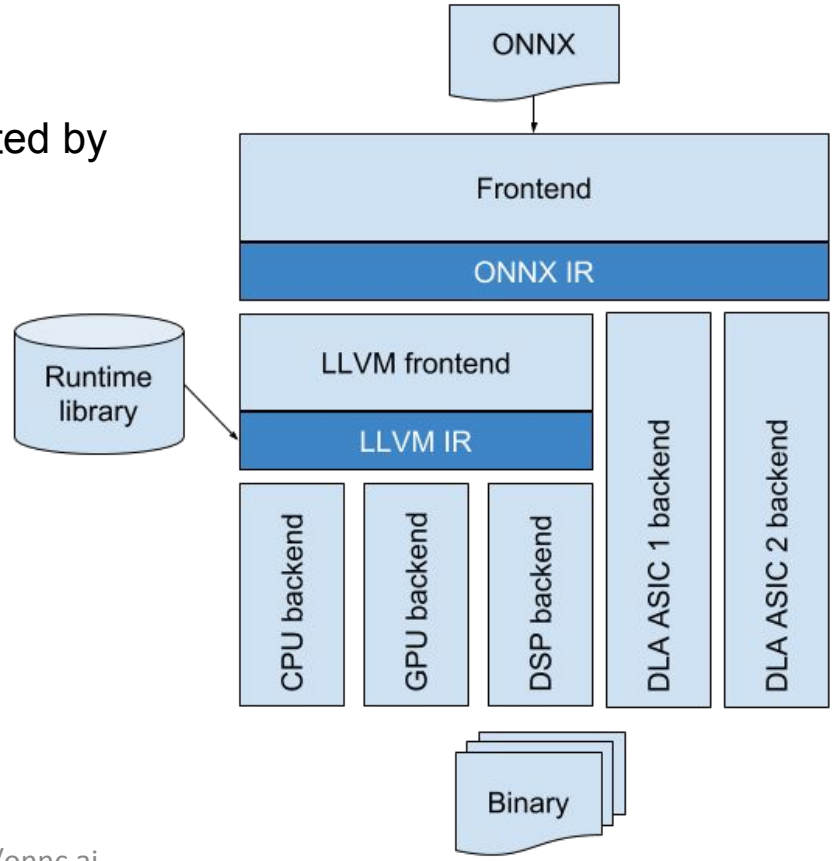


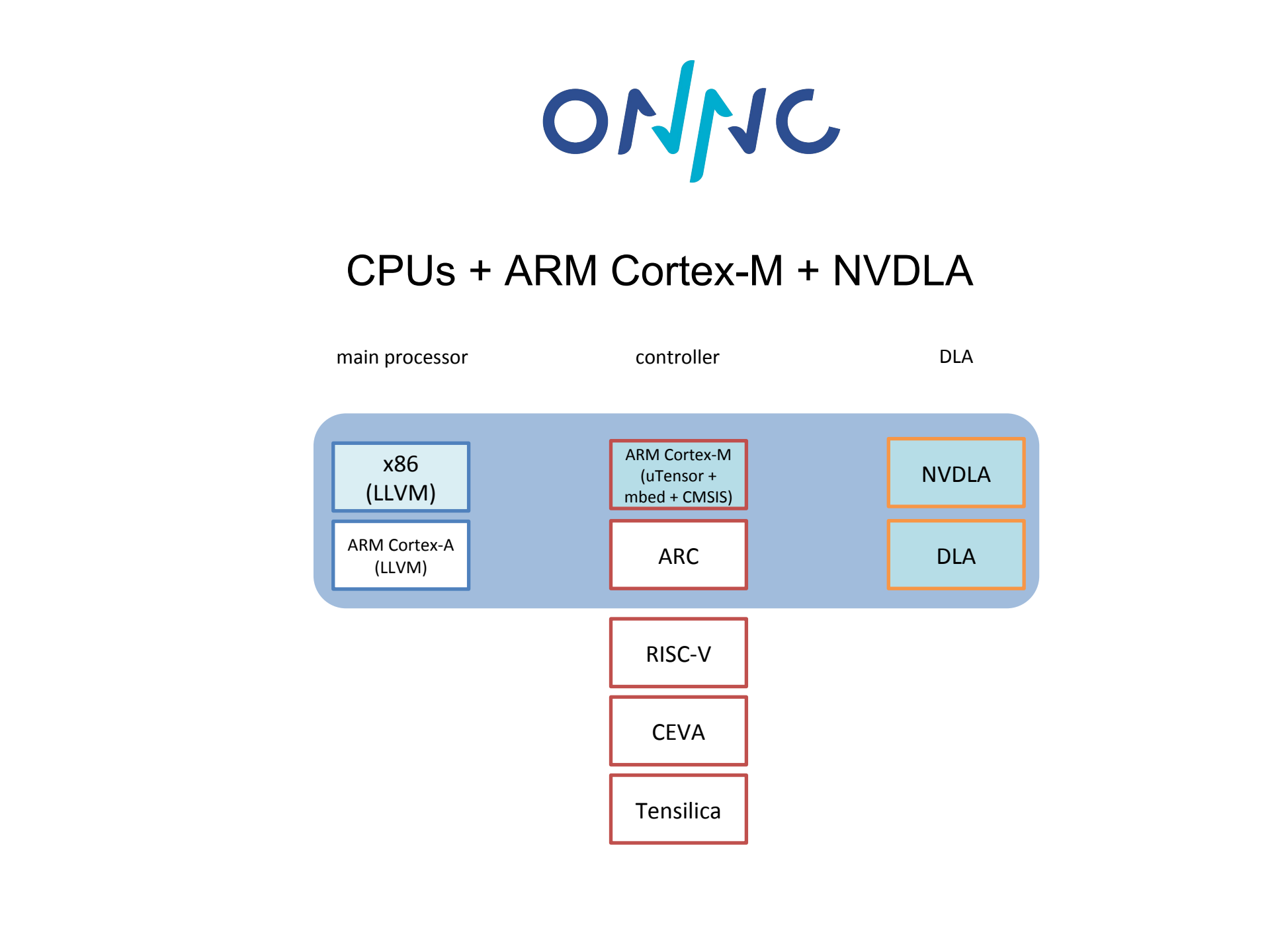
Add D, PassManager will add A and B automatically

Connect to both

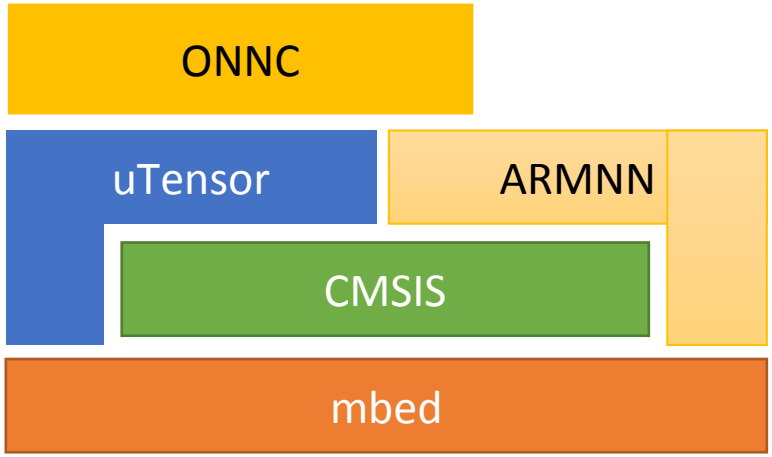
LLVM and ASIC

- No porting effort for LLVM compiler
- Support ASIC which can NOT be supported by LLVM



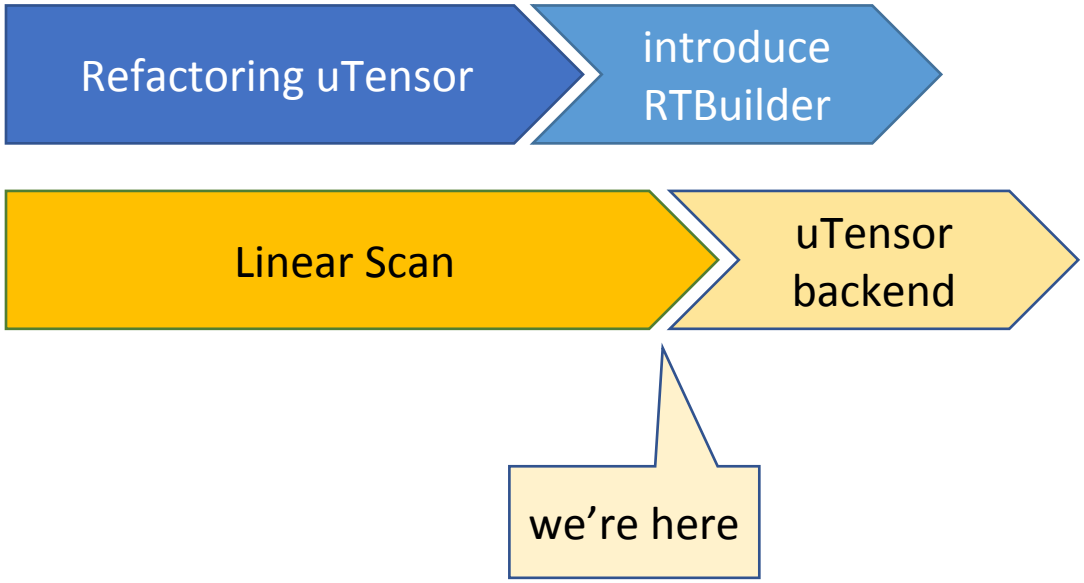


CPUs + ARM Cortex-M + NVDLA



changes for uTensor

changes in ONNC



Projects reside in

https://repo.onnc.ai

The Regression project for testers

for developers

The Umbrella project for porters

Current Status

0.9.2

Next release

0.9.3

Version	release date	Summary
0.9.0	8/2	Initial release
0.9.1	8/3	Sophon DLA
0.9.2	9/18	memory allocation (linear scan algorithm)
0.9.3	10/1 (ETA)	x86 interpreter Platform registry
1.0.0	10/14 (ETA)	x86 JIT and bundle uTensor/CMSIS interpreter
1.1.0	10/31 (ETA)	NVDLA bundle Platform passes

Search or jump to... / Pull requests Issues Marketplace Explore

ONNC / onnc

Watch 12 Star 11 Fork 2

Code Issues 0 Pull requests 1 Projects 0 Wiki Insights Settings

Open Neural Network Compiler https://onnc.ai

Add topics

Edit

Give me food and **Stars**, please

ONNC

https://repo.onnc.ai

https://onnc.ai