NN Synthesis Project

吳育丞 吳冠緯

Outline

- 1. Model Training
  - Data Augmentation
  - ReLU Upper Bound
  - Hyper Parameter (lr, batch size, optimizer ...)
  - Knowledge Distillation
- 2. Circuit Generation
  - Generate Matrix
  - Find Sharing
  - Construct Graph & MaxWeightMatching
- 3. FPGA Implementation (Bonus)
  - Simulation
  - Synthesis
  - Implementation

1. Model Training

Data Augmentation

transforms.ToTensor(),

transforms.RandomRotation(5),

transforms.RandomResizedCrop(28),

transforms.Normalized((0.5,), (0.5,))

ReLU Upper Bound

torch.nn.functional.relu6() → limit upper bound to 6

[ReLU6(x) = min(max(0, x), 6)]

torch.nn.functional.relu() + min(x, 7) → limit upper bound to 7

4 bits are used on both upper bounds

Hyper Parameter

Default:

bit = 4

lr = 0.0005 (0.005, 0.00005 ...)

dropout = 0.35 (0.5 ...)

batch size = 256 (128 ...)

epoch = 100 (150 ...)

optimizer = Adam (RAdam, SGD ...)

Large Model Example

c1 (1, 3, k=3, s=2, p=0)

f1 (3*13*13, 64)

f2 (64, 32)

f3 (32, 10)

```
Testing accuracy of the quantized model:  0.9701
```

```
--- Number of parameter ---
34958
epoch:5, train acc:0.9141, valid acc:0.9061
epoch:10, train acc:0.94134, valid acc:0.9355
epoch:15, train acc:0.95994, valid acc:0.9555
epoch:20, train acc:0.96434, valid acc:0.9565
epoch:25, train acc:0.96898, valid acc:0.9603
epoch:30, train acc:0.97014, valid acc:0.964
epoch:35, train acc:0.9699, valid acc:0.9617
epoch:40, train acc:0.97166, valid acc:0.9626
epoch:45, train acc:0.97664, valid acc:0.9679
epoch:50, train acc:0.97474, valid acc:0.9666
epoch:55, train acc:0.97512, valid acc:0.9682
epoch:60, train acc:0.97536, valid acc:0.9666
epoch:65, train acc:0.97206, valid acc:0.9648
epoch:70, train acc:0.9756, valid acc:0.9683
epoch:75, train acc:0.97658, valid acc:0.9676
epoch:80, train acc:0.97002, valid acc:0.9627
epoch:85, train acc:0.97458, valid acc:0.965
epoch:90, train acc:0.97388, valid acc:0.966
epoch:95, train acc:0.97612, valid acc:0.9686
epoch:100, train acc:0.9746, valid acc:0.9652
best valid acc: 0.9686
```

Small Model Example

c1 (1, 3, k=11, s=3, p=0)

f1 (3*6*6, 10)

```
Testing accuracy of the quantized model:  0.953
```

```
--- Number of parameter ---
1462
epoch:5, train acc:0.90762, valid acc:0.9062
epoch:10, train acc:0.92722, valid acc:0.9291
epoch:15, train acc:0.9338, valid acc:0.9311
epoch:20, train acc:0.94408, valid acc:0.9421
epoch:25, train acc:0.94722, valid acc:0.9465
epoch:30, train acc:0.94766, valid acc:0.9452
epoch:35, train acc:0.94636, valid acc:0.9446
epoch:40, train acc:0.95076, valid acc:0.9447
epoch:45, train acc:0.9503, valid acc:0.95
epoch:50, train acc:0.93756, valid acc:0.9405
epoch:55, train acc:0.94084, valid acc:0.9386
epoch:60, train acc:0.94088, valid acc:0.9404
epoch:65, train acc:0.93826, valid acc:0.9335
epoch:70, train acc:0.93, valid acc:0.9316
epoch:75, train acc:0.94516, valid acc:0.9457
epoch:80, train acc:0.94968, valid acc:0.9442
epoch:85, train acc:0.94236, valid acc:0.9394
epoch:90, train acc:0.92982, valid acc:0.9312
epoch:95, train acc:0.94402, valid acc:0.9409
epoch:100, train acc:0.9434, valid acc:0.9421
best valid acc: 0.95
```

Different Modification: Model Architecture

fully connected NN model (4 layers):

256 / 128 / 64 / 10 : 98%

128 / 64 / 32 / 10 : 97%

64 / 48 / 16 / 10 : 97%

48 / 32 / 16 / 10 : 95%

Note that for 2 layers 256 / 10 can obtain 97% acc as well.

Different Modification: lr

change the learning rate of Adam optimizer:

batch size = 256

bit = 4

	lr = 0.005	lr = 0.0005	lr = 0.00005
best valid	0.9356	0.9566	0.9529
test accuracy	0.9347	0.9547	0.9538

Different Modification: bit

change the learning rate of Adam optimizer:

batch size = 256

lr = 0.0005

	bit = 4	bit = 6	bit = 8
best valid	0.9566	0.9776	0.9777
test accuracy	0.9547	0.9764	0.9764

Different Modification: batch size

change the learning rate of Adam optimizer:

bit = 4

lr = 0.0005

	size = 256	size = 128	size = 64
best valid	0.9566	0.9488	0.953
test accuracy	0.9574	0.9512	0.9593

Different Modification: ReLU upper bound

change the learning rate of Adam optimizer:

bit = 4

lr = 0.0005

batch size = 256

	limit = 5	limit = 6 (relu6)	limit = 7
best valid	0.9575	0.9628 (0.9566)	0.9536
test accuracy	0.9618	0.9634 (0.9574)	0.9535

Different Modification: bit quantization

Quantizing activations (a) and weights (w) to 4-bit and 3-bit

default (case I): conv (1, 3, 3, 2) / fc (3*13*13, 32, 32, 10)

case II: fc (3*13*13, 80, 32, 10) / case III: conv (1, 5, 3, 2)

	a4 w4 (I)	a4 w3 (I)	a3 w3 (I)	a3 w3 (II)	a3 w3 (III)
best valid	95.99%	96.22%	93.86%	94.92%	95.68%
test accu.	96.21%	96.21%	93.61%	95.42%	95.32%
AIG accu.	95.73%	86.93%	84.8%	95.1%	93.77%
AND gates #	516492	309698	245716	323941	325484

Different Modification: Knowledge Distillation

lr = 0.002

batch size = 256

teacher model: conv (1, 8, 3, 2, 0) / fc (512, 128, 10)

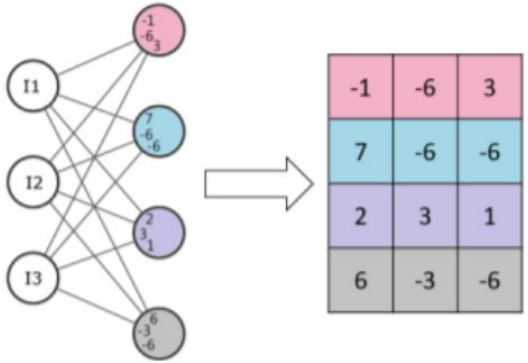
use learning rate schedule: ExponentialLR with gamma = 0.95

without KD (with KD)	4-bit teacher	4-bit student	3-bit teacher	3-bit student (**α**=0.9)	3-bit student (**α**=0.9-0.1)
best valid	98.56	96.24(95.66)	98	95.15(93.86)	94.08(93.86)
test accu.	98.36	96.12(95.47)	97.91	95.06(93.61)	94.49(93.61)

2. Circuit Generation

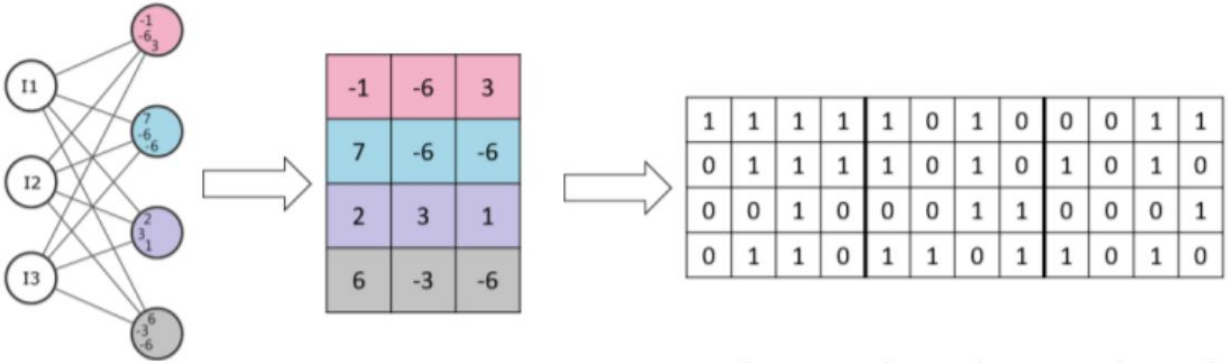
Y.-S. Huang, J.-H. R. Jiang, and A. Mishchenko,
"Quantized neural network synthesis for direct logic circuit implementation", Proc. IWLS'21.

Generate Matrix



$$O_1 = -I_1 - 6I_2 + 3I_3$$
$$O_2 = 7I_1 - 6I_2 - 6I_3$$
$$O_3 = 2I_1 + 3I_2 + I_3$$
$$O_4 = 6I_1 - 3I_2 - 6I_3$$

Generate Matrix



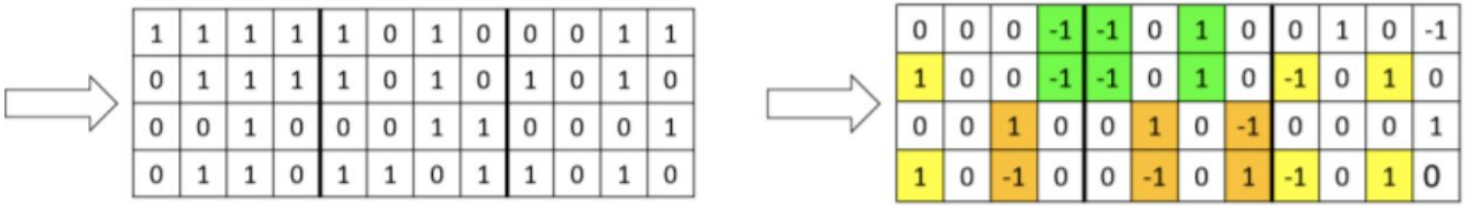
$$O_1 = -(I_1 \ll 3) + (I_1 \ll 2) + (I_1 \ll 1) + I_1 - (I_2 \ll 3) + (I_2 \ll 1) + (I_3 \ll 1) + I_3$$

$$O_2 = (I_1 \ll 2) + (I_1 \ll 1) + I_1 - (I_2 \ll 3) + (I_2 \ll 1) - (I_3 \ll 3) + (I_3 \ll 1)$$

$$O_3 = (I_1 \ll 1) + (I_2 \ll 1) + I_2 + I_3$$

$$O_4 = (I_1 \ll 2) + (I_1 \ll 1) - (I_2 \ll 3) + (I_2 \ll 2) + I_2 - (I_3 \ll 3) + (I_3 \ll 1)$$

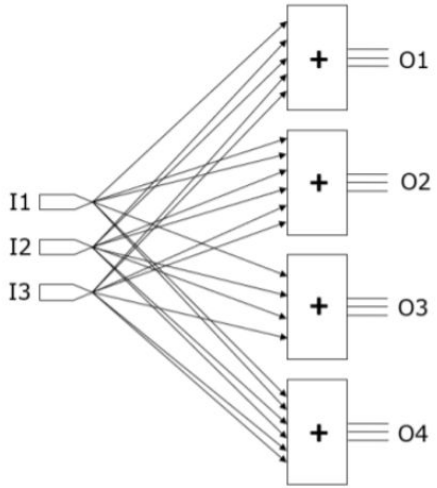
Booth Matrix



$$O_1 = -I_1 - (I_2 \ll 3) + (I_2 \ll 1) + (I_3 \ll 2) - I_3$$
$$O_2 = (I_1 \ll 3) - I_1 - (I_2 \ll 3) + (I_2 \ll 1) - (I_3 \ll 3) + (I_3 \ll 1)$$
$$O_3 = (I_1 \ll 1) + (I_2 \ll 2) - I_2 + I_3$$
$$O_4 = (I_1 \ll 3) - (I_1 \ll 1) - (I_2 \ll 2) + I_2 - (I_3 \ll 3) + (I_3 \ll 1)$$

Find Sharing

0	0	0	-1	-1	0	1	0	0	1	0	-1
1	0	0	-1	-1	0	1	0	-1	0	1	0
0	0	1	0	0	1	0	-1	0	0	0	1
1	0	-1	0	0	-1	0	1	-1	0	1	0

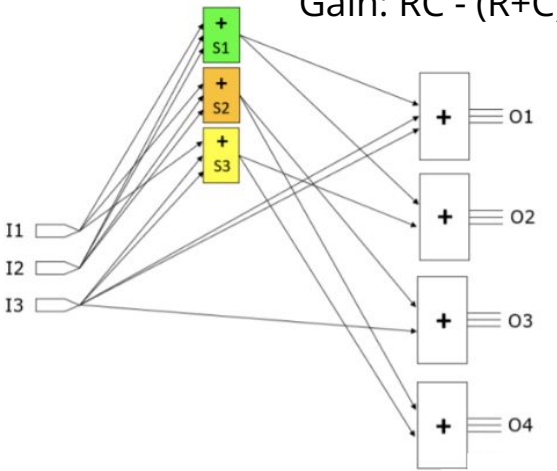


$$O_1 = -I_1 - (I_2 \ll 3) + (I_2 \ll 1) + (I_3 \ll 2) - I_3$$
$$O_2 = (I_1 \ll 3) - I_1 - (I_2 \ll 3) + (I_2 \ll 1) - (I_3 \ll 3) + (I_3 \ll 1)$$
$$O_3 = (I_1 \ll 1) + (I_2 \ll 2) - I_2 + I_3$$
$$O_4 = (I_1 \ll 3) - (I_1 \ll 1) - (I_2 \ll 2) + I_2 - (I_3 \ll 3) + (I_3 \ll 1)$$

Find Sharing

0	0	0	-1	-1	0	1	0	0	1	0	-1
1	0	0	-1	-1	0	1	0	-1	0	1	0
0	0	1	0	0	1	0	-1	0	0	0	1
1	0	-1	0	0	-1	0	1	-1	0	1	0

Gain: RC - (R+C)



$$O_1 = -I_1 - (I_2 \ll 3) + (I_2 \ll 1) + (I_3 \ll 2) - I_3$$
$$O_2 = (I_1 \ll 3) - I_1 - (I_2 \ll 3) + (I_2 \ll 1) - (I_3 \ll 3) + (I_3 \ll 1)$$
$$O_3 = (I_1 \ll 1) + (I_2 \ll 2) - I_2 + I_3$$
$$O_4 = (I_1 \ll 3) - (I_1 \ll 1) - (I_2 \ll 2) + I_2 - (I_3 \ll 3) + (I_3 \ll 1)$$

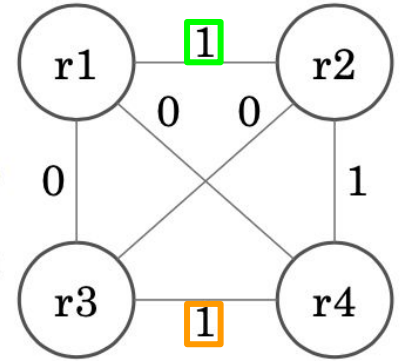
Find Sharing

Algorithm **FindMaxPairing** (W_B, *unSharedCols*)
  $G(V, E)$ = **ConstructGraph** (W_B, *unSharedCols*)
  pairs = **MaxWeightMatching**(V, E)
  result = [ ]
  foreach *pair* in *pairs*:
    S^{same}, S^{diff} = **PairRows** (*pair*)
    if **gain** (S^{same}) > 0:
      append S^{same} to *result*
    if **gain** (S^{diff}) > 0:
      append S^{diff} to *result*
return *result*

Construct Graph

0	0	0	-1	-1	0	1	0	0	1	0	-1
1	0	0	-1	-1	0	1	0	-1	0	1	0
0	0	1	0	0	1	0	-1	0	0	0	1
1	0	-1	0	0	-1	0	1	-1	0	1	0

ConstructGraph



gain(S_{same}) + gain(S_{diff})

MaxWeightMatching

Z. Galil, “Efficient algorithms for finding maximum matching in graphs,” *ACM Comput. Surv.*, vol. 18, no. 1, pp. 23–38, 1986.

Overall Algorithm

Algorithm 1 FindSharing

Input: An $m \times nb$ Booth matrix W_B
Output: A list of shared terms

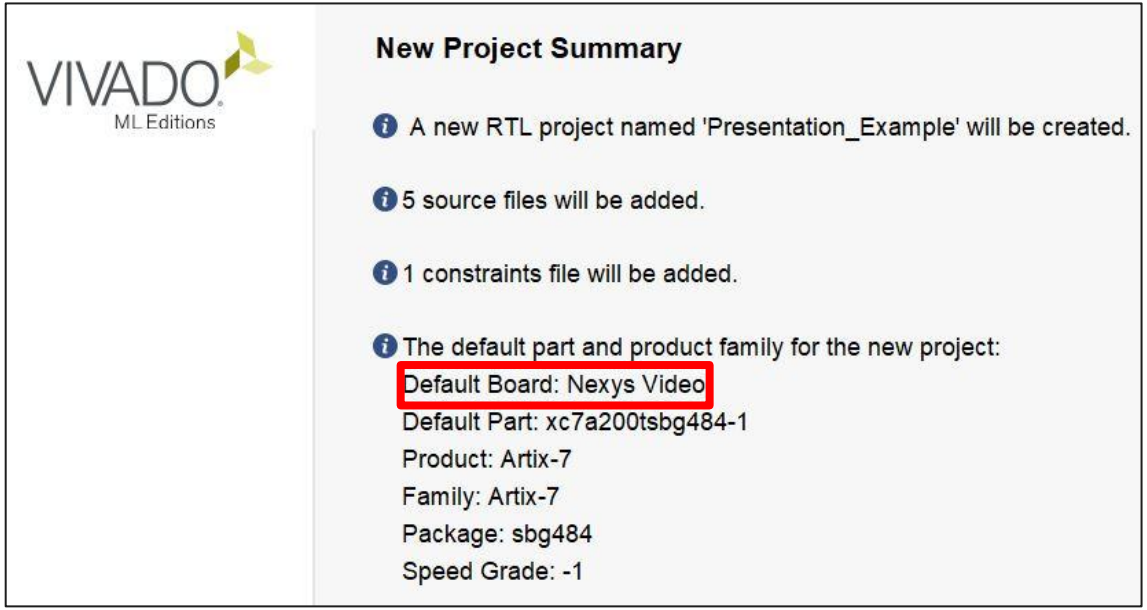
1: $C := \{0, \ldots, nb-1\}$;
2: *unSharedCols* := an empty hash table;
3: **for** $r = 0$ to $m-1$ **do**
4: *unSharedCols*[r] = C;
5: *result* := [];
6: *sharings* := *FindMaxPairing*(W_B,*unSharedCols*);
7: **while** *size*(*sharings*)> 0 **do**
8: append elements in *sharings* to *result*;
9: *unSharedCols* := *refine*(*unSharedCols*,*sharings*);
10: *sharings* := *FindMaxPairing*(W_B,*unSharedCols*);
11: **return** *result*;

3. FPGA Implementation (Bonus)

Example Result

c1 (1, 2, k=3, s=2, p=0)

f1 (2*13*13, 10)



Simulation

```
pred:0000100000
correct
========
correct count:        9285
========
index:        9999 true label:  6

pred:0001000000
correct
========
correct count:        9286
$finish called at time : 1400105 ns : File "C:/Users/User/Desktop/eda_project/W3/lab/files/sim/tb_unpacked.sv" Line 59
run: Time (s): cpu = 00:00:16 ; elapsed = 00:00:12 . Memory (MB): peak = 1395.102 ; gain = 36.078
```

Synthesis: Time

synth_1 (active)	constrs_1	synth_design Complete!								69184	6716	0.0
impl_1	constrs_1	Not started										
Out-of-Context Module Runs												
blk_mem_gen_0_synth_1	blk_mem_gen_0	synth_design Complete!								0	0	174.5

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	3.377 ns	Worst Hold Slack (WHS):	0.139 ns	Worst Pulse Width Slack (WPWS):	9.500
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	15368	Total Number of Endpoints:	15368	Total Number of Endpoints:	7067

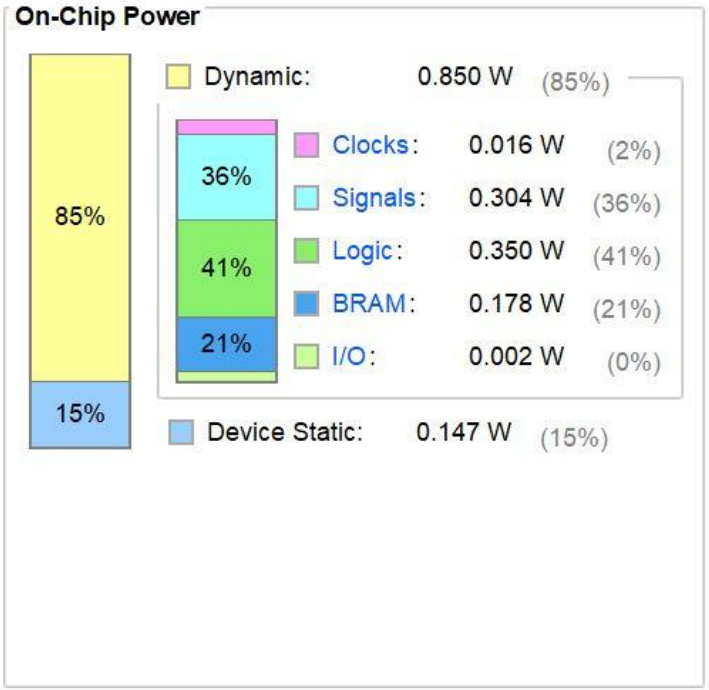
All user specified timing constraints are met.

Synthesis: Power

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:	**0.998 W**
Design Power Budget:	**Not Specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**28.3°C**
Thermal Margin:	56.7°C (16.7 W)
Effective ϑJA:	3.3°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Medium

Launch Power Constraint Advisor to find and fix invalid switching activity



Implementation: Time

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF
synth_1 (active)	**constrs_1**	**synth_design Complete!**								**69184**	**6716**
impl_1	constrs_1	route_design Complete!	0.471	0.000	0.066	0.000	0.000	0.659	0	57876	6717
Out-of-Context Module Runs											
blk_mem_gen_0_synth_1	blk_mem_gen_0	synth_design Complete!								0	0

BRAM	URAM	DSP	Start	Elapsed	Run Strategy	Report Strategy
0.0	**0**	**0**	**6/16/22, 11:27 PM**	**00:14:51**	**Flow_AlternateRoutability (Vivado Synthesis 2021)**	**Vivado Synthesis Default Reports (Vivado Synthesis 202**
174.5	0	0	6/16/22, 11:44 PM	00:29:06	Congestion_SpreadLogic_medium (Vivado Implementation 2021)	Vivado Implementation Default Reports (Vivado Implementat
174.5	0	0	6/16/22, 11:19 PM	00:05:32	Vivado Synthesis Defaults (Vivado Synthesis 2021)	Vivado Synthesis Default Reports (Vivado Synthesis 2021)

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	0.471 ns	Worst Hold Slack (WHS):	0.066 ns	Worst Pulse Width Slack (WPWS):	9.500 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	15719	Total Number of Endpoints:	15719	Total Number of Endpoints:	7068

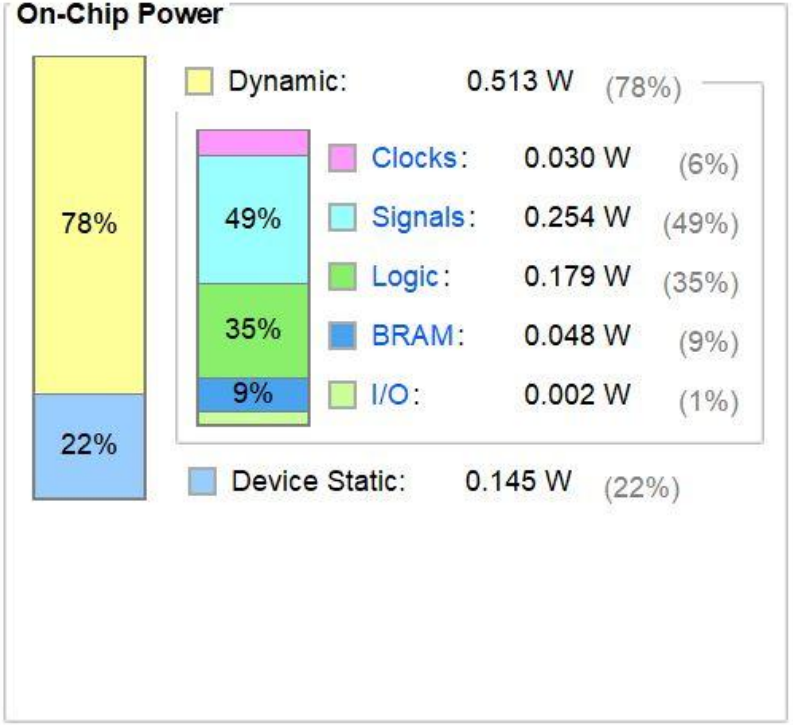
All user specified timing constraints are met.

Implementation: Power

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:	**0.659 W**
Design Power Budget:	**Not Specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**27.2°C**
Thermal Margin:	57.8°C (17.1 W)
Effective ϑJA:	3.3°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Medium

Launch Power Constraint Advisor to find and fix invalid switching activity



Q&A