Efficient Weight Reuse for Large LSTMs

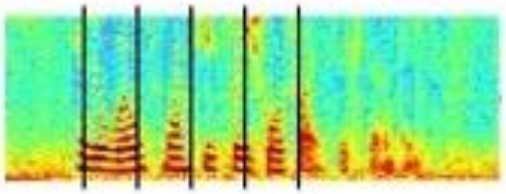
Zhiqiang Que[1], Thomas Nugent[1], Shuanglong Liu[1], Li Tian[3], Xinyu Niu[2], Yongxin Zhu[3], Wayne Luk[1]

[1]Imperial College London, [2]Corerain Technologies Ltd. , [3]Chinese Academy of Science

Outline

- **Motivation**
- **Design & Implementation**
  - **Stall-free hardware architecture**
  - **Blocking-batching strategy**
  - **SBE FPGA Accelerator**
- **Evaluation**
  - **Batch size and Blocking number**
  - **Performance and Efficiency**
- **Future Work and Summary**

Speech Recognition



Acoustic Input

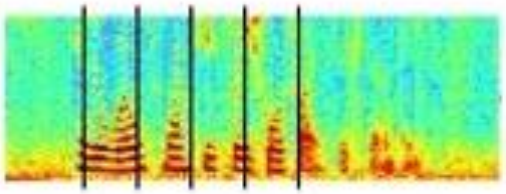
Deep Recurrent Neural Network

"How cold is it outside?"

Text Output

Google Research Blog, August 2015

Speech Recognition



Acoustic Input

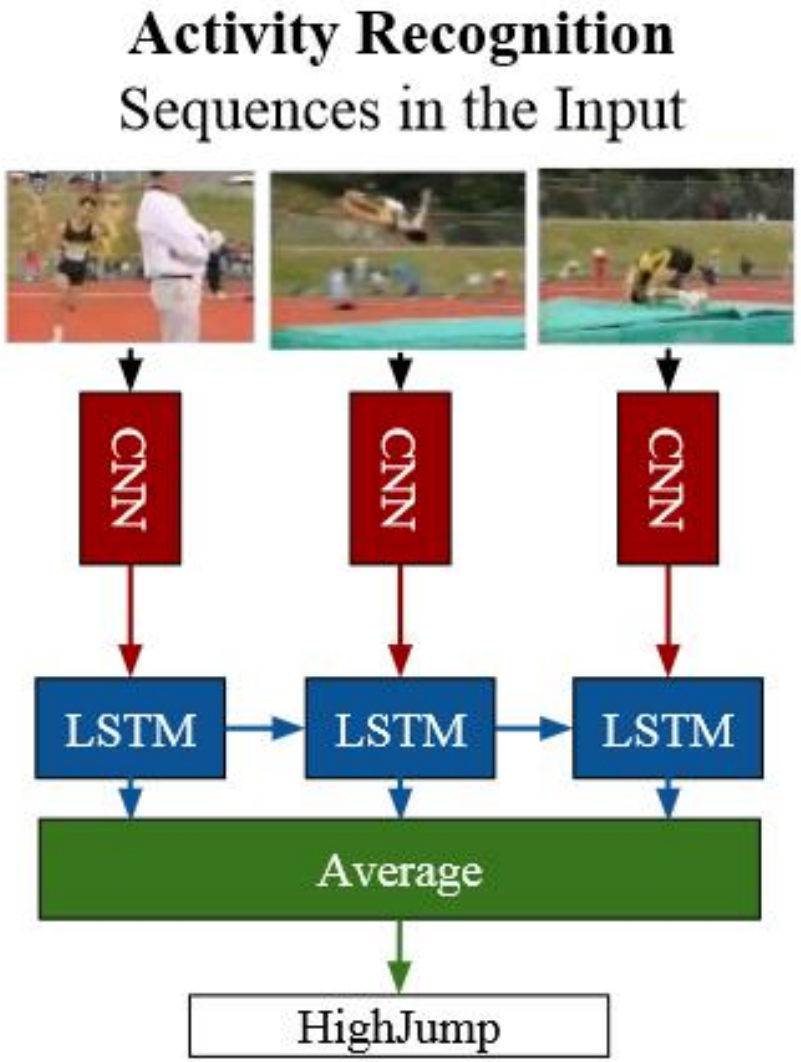
Deep Recurrent Neural Network

"How cold is it outside?"

Text Output

Google Research Blog, August 2015

Video Analysing



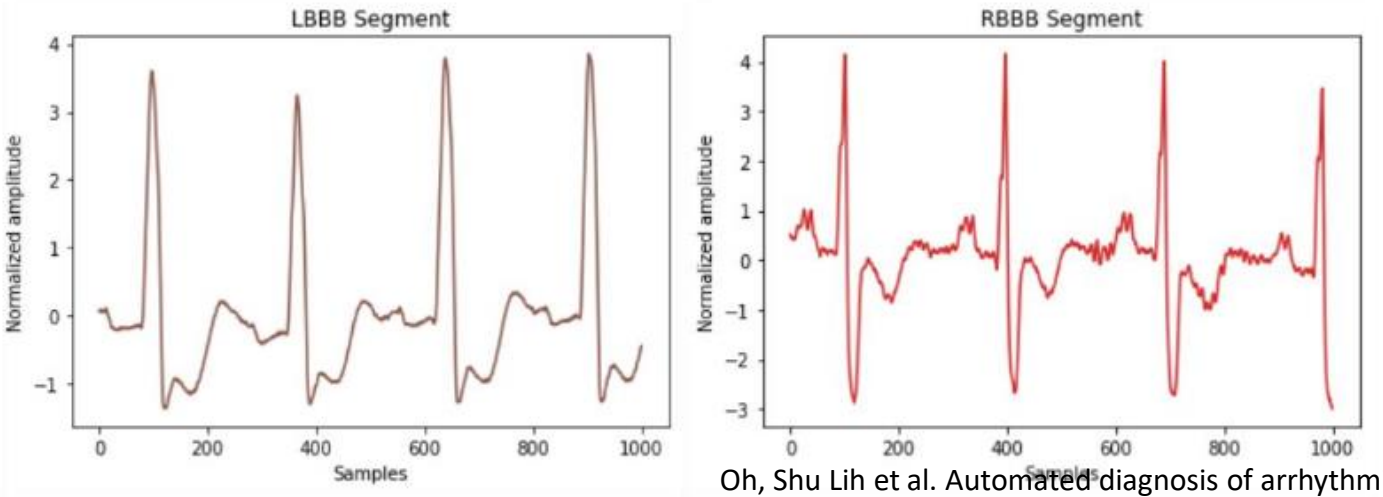
J. Donahue et al. "LRCN"

- COMAC:

on-board flight testing

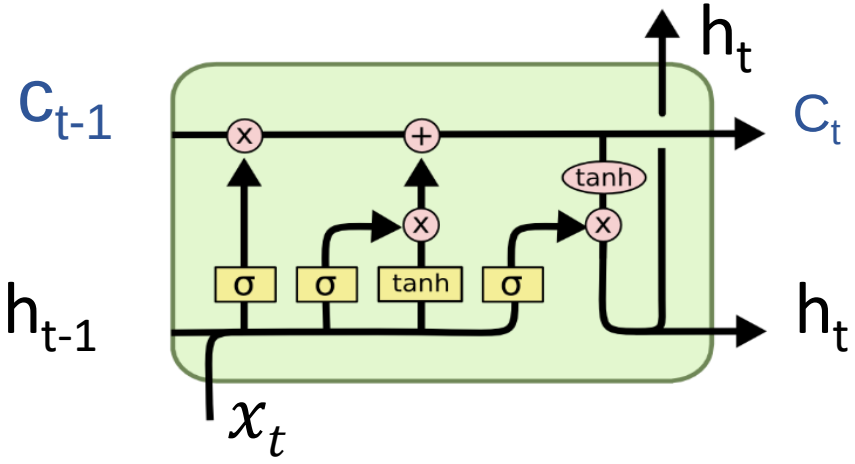
•COMAC:
on-board flight testing

- ECG Anomaly Detection



Oh, Shu Lih et al. Automated diagnosis of arrhythmia..

LSTM Overview



$$i_t = \sigma(W_i[x_t, h_{t-1}] + b_i)$$
$$f_t = \sigma(W_f[x_t, h_{t-1}] + b_f)$$
$$u_t = \sigma(W_u[x_t, h_{t-1}] + b_u)$$
$$o_t = \sigma(W_o[x_t, h_{t-1}] + b_o)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot u_t$$
$$h_t = o_t \odot tanh(c_t)$$

Most of the calculations within LSTM cells lie in the Matrix-Vector Multiplication (MV)

Motivation: Challenges

- C1: RNN Data dependence: stall in execution
- C2: Large RNN: weights stored on external DRAM

Contributions

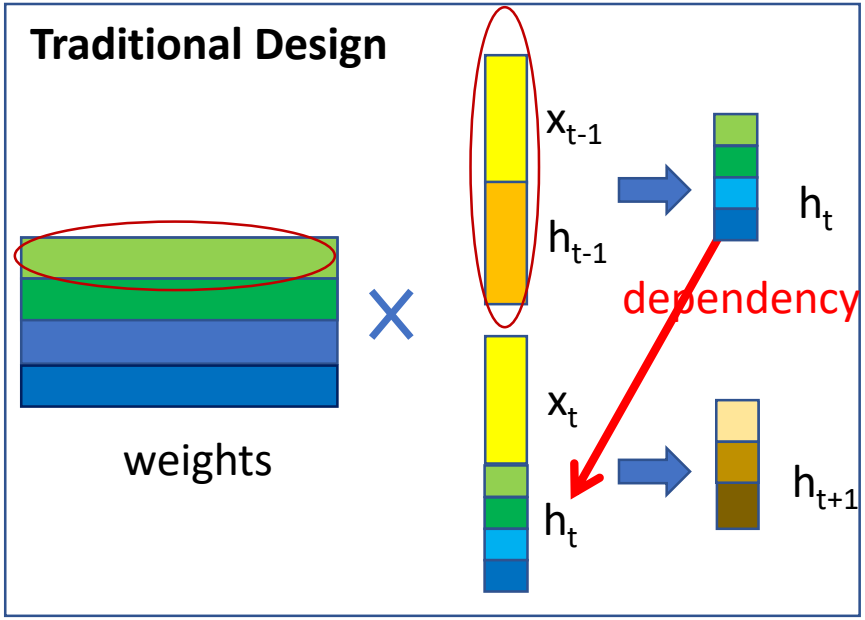
- Blocking-batching strategy (address C2)
  - reuse weights for large LSTM systems
- Stall-free architecture (address C1)
  - reorganise multiplications to enhance throughput
- Zynq and Virtex-7 implementations
  - 23.7x speed, 1/208x energy of CPU
  - 1.3x speed, 1/19.2x energy of GPU

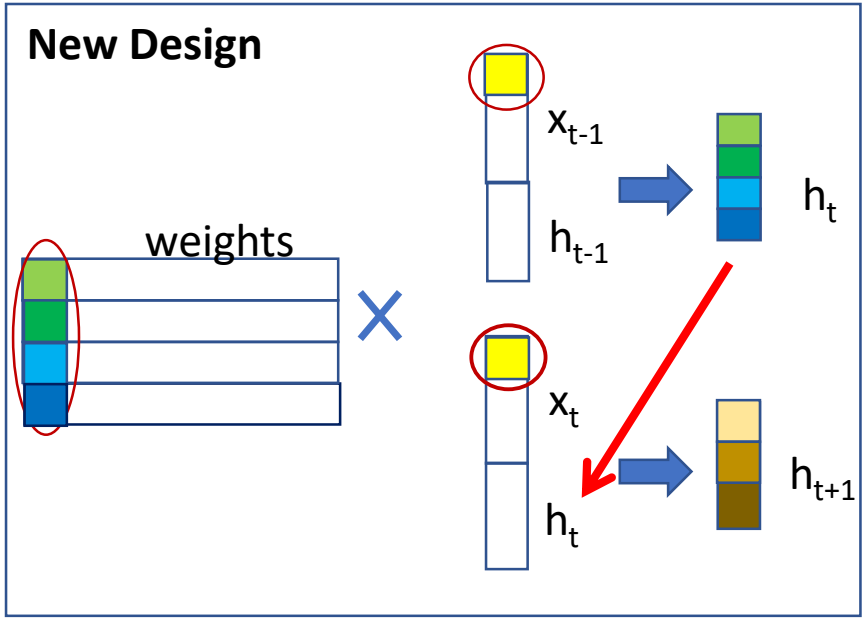
Outline

- Motivation
- **Design & Implementation**
  - **Stall-free hardware architecture**
  - Blocking-batching strategy
  - SBE FPGA Accelerator
- Evaluation
  - Batch size and Blocking number
  - Performance and Efficiency
- Future Work and Summary

Stall-free hardware architecture

For simplicity,
Only one element is involved





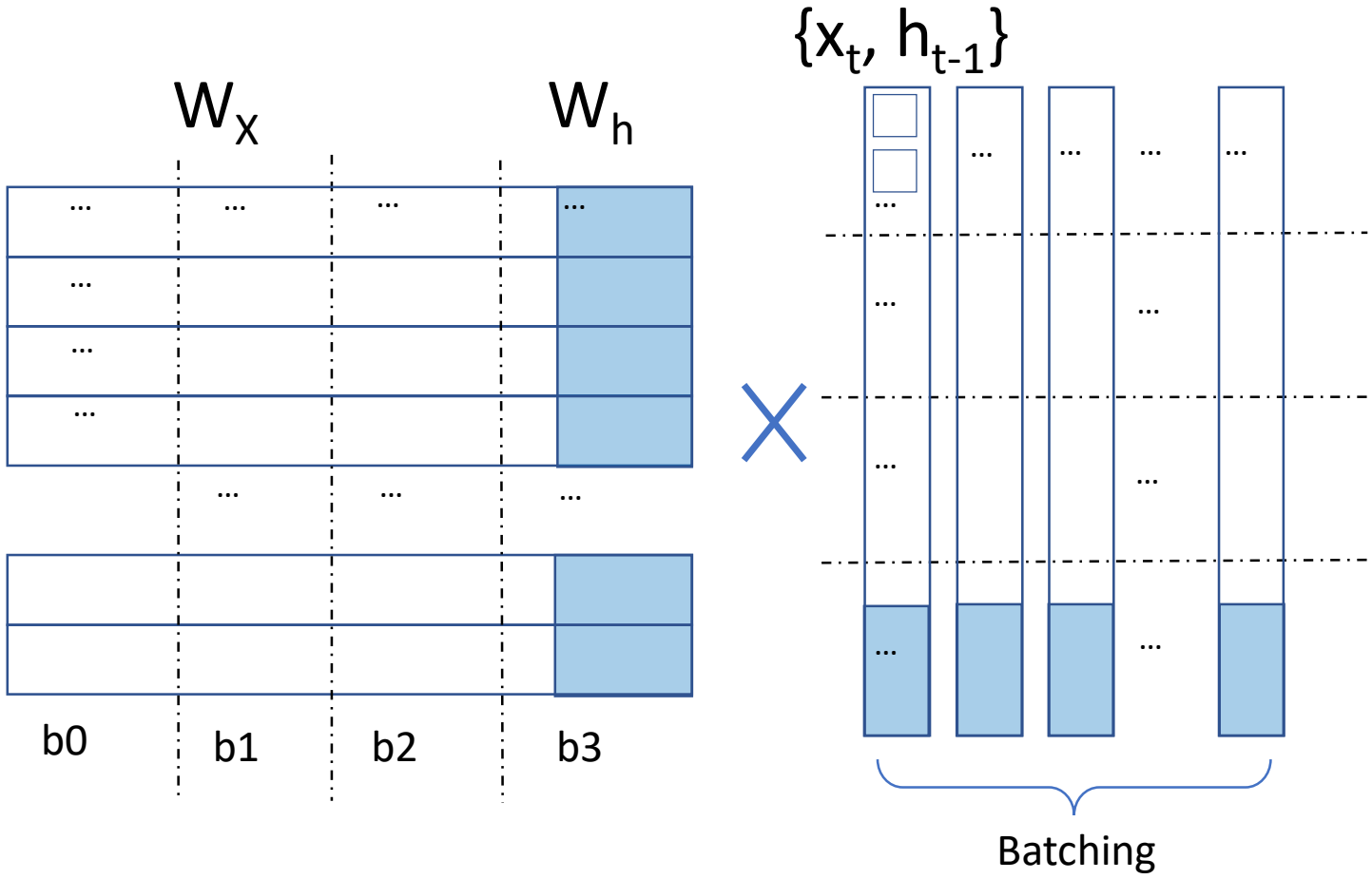

```
M x V :    for i < height:
               for j < width:
                   r[i] += w[i][j] * x[j]
```

```
New M x V :    for j < width:
                   for i < height:
                       r[i] += w[i][j] * x[j]
```

Outline

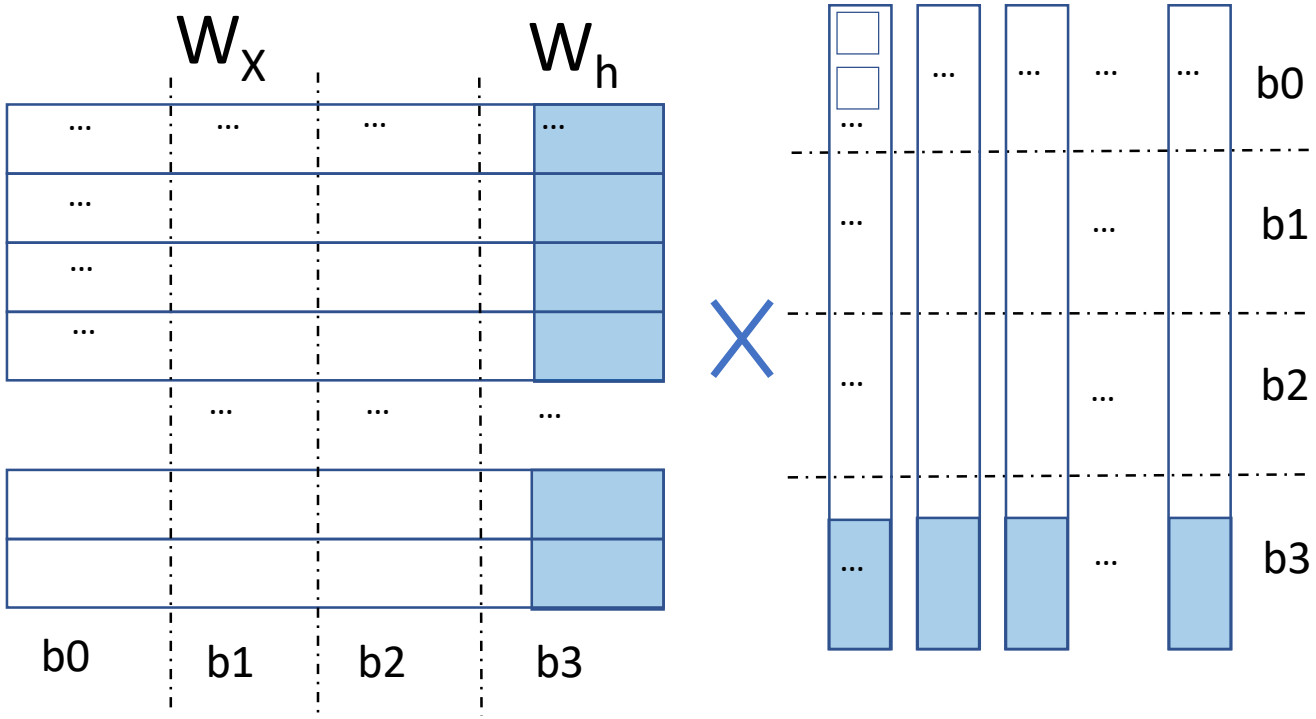
- Motivation
- **Design & Implementation**
  - Stall-free hardware architecture
  - **Blocking-batching strategy**
  - SBE FPGA Accelerator
- Evaluation
  - Batch size and Blocking number
  - Performance and Efficiency
- Future Work and Summary

Blocking and Batching



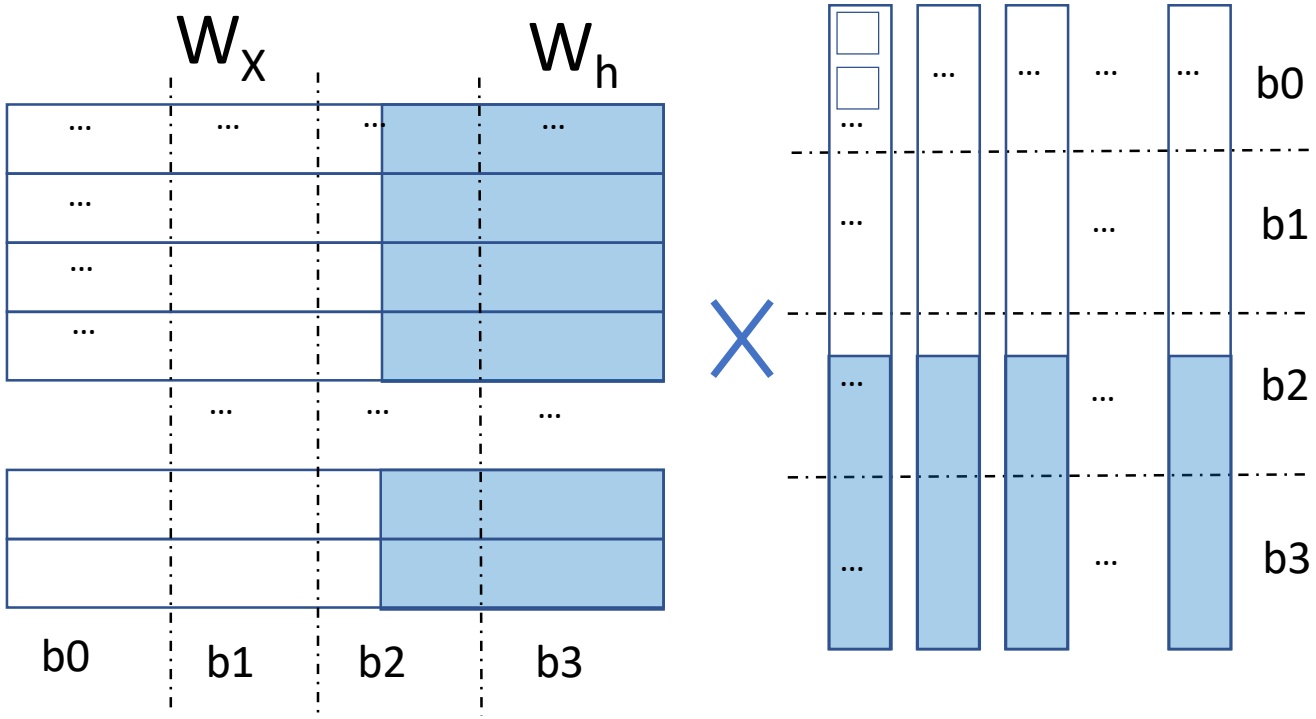
3 Scenarios- Case1

1. The hidden unit weights can be stored in one block



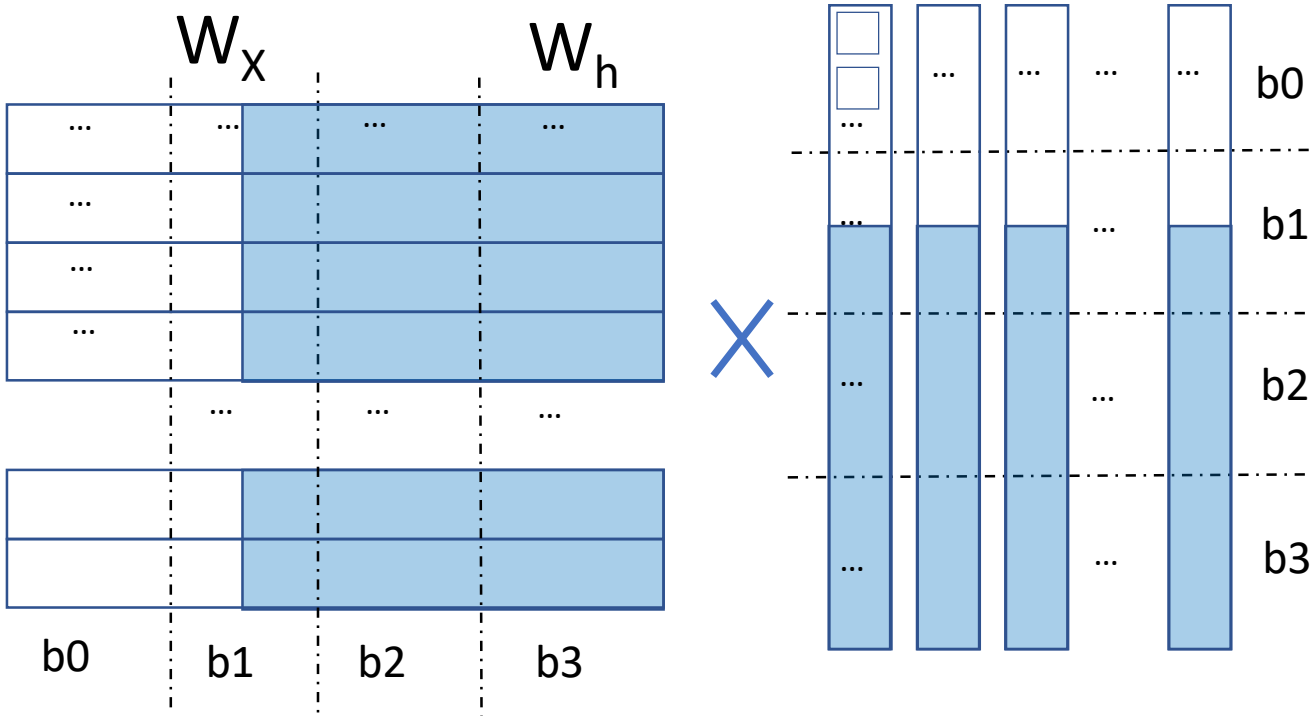
3 Scenarios- Case2

2. The hidden unit weights can be stored in two blocks



3 Scenarios- Case3

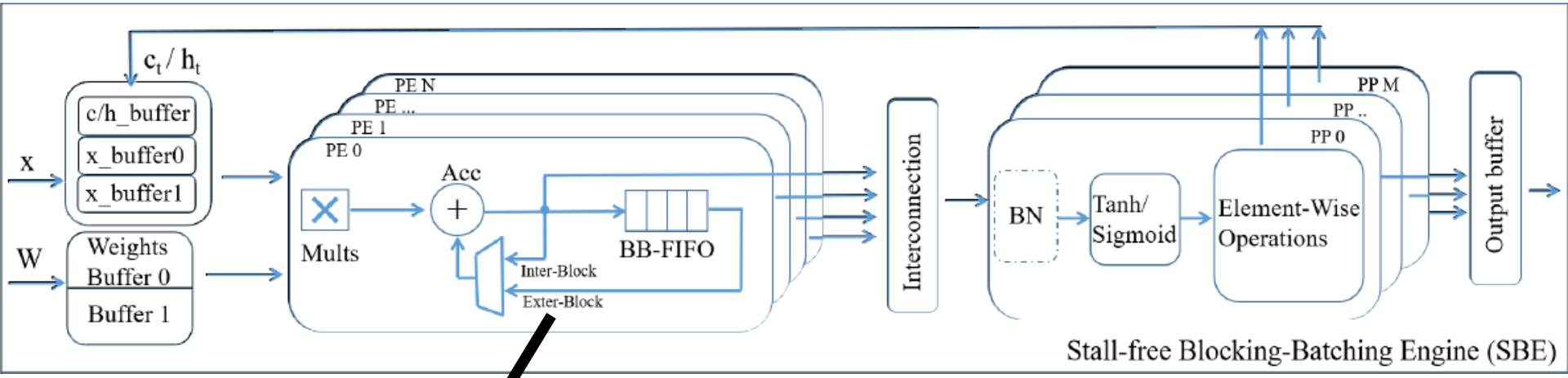
3. The hidden unit weights can be stored in more than two blocks

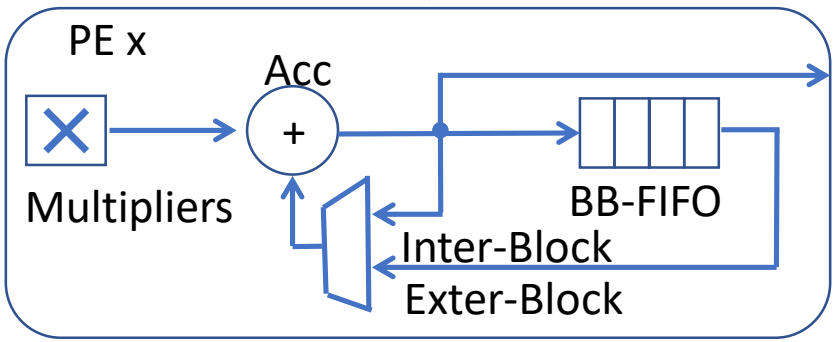


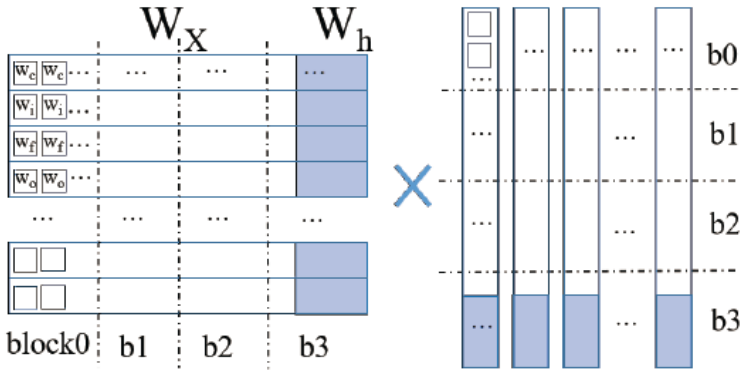
Outline

- Motivation
- **Design & Implementation**
  - Stall-free hardware architecture
  - Blocking-batching strategy
  - **SBE FPGA Accelerator**
- Evaluation
  - Batch size and Blocking number
  - Performance and Efficiency
- Future Work and Summary

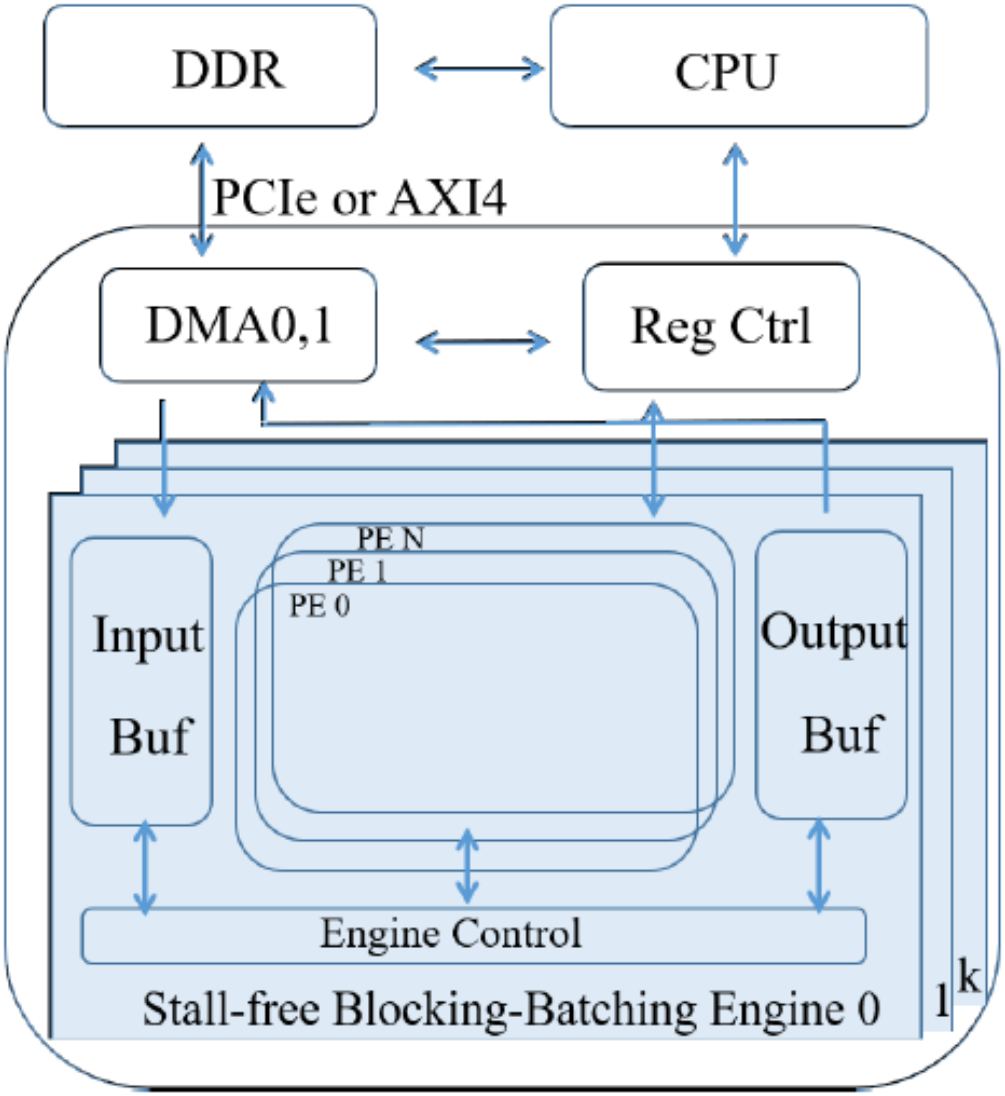
SBE Architecture Details







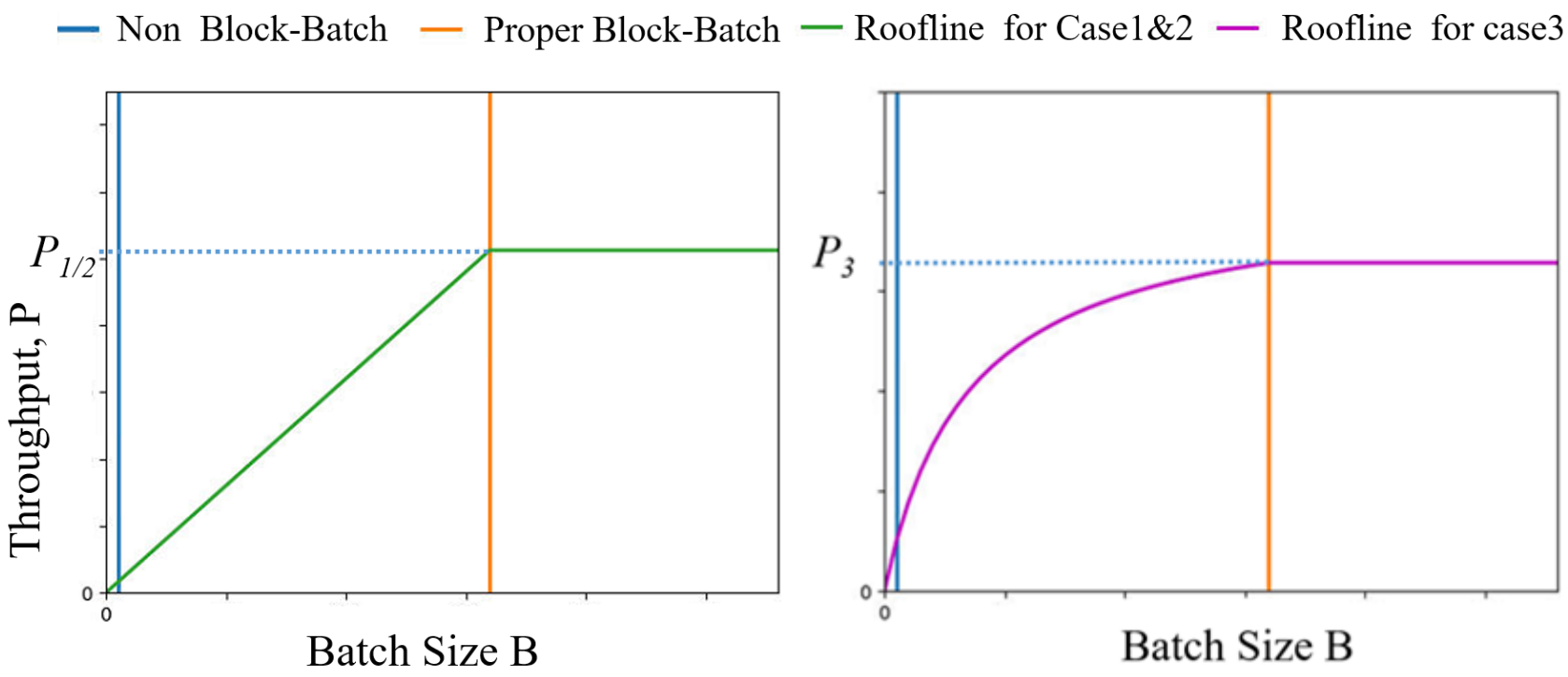
System Overview



Outline

- Motivation
- Design & Implementation
  - Stall-free hardware architecture
  - Blocking-batching strategy
  - SBE FPGA Accelerator
- **Evaluation**
  - **Batch size and Blocking number**
  - Performance and Efficiency
- Future Work and Summary

Batching Size



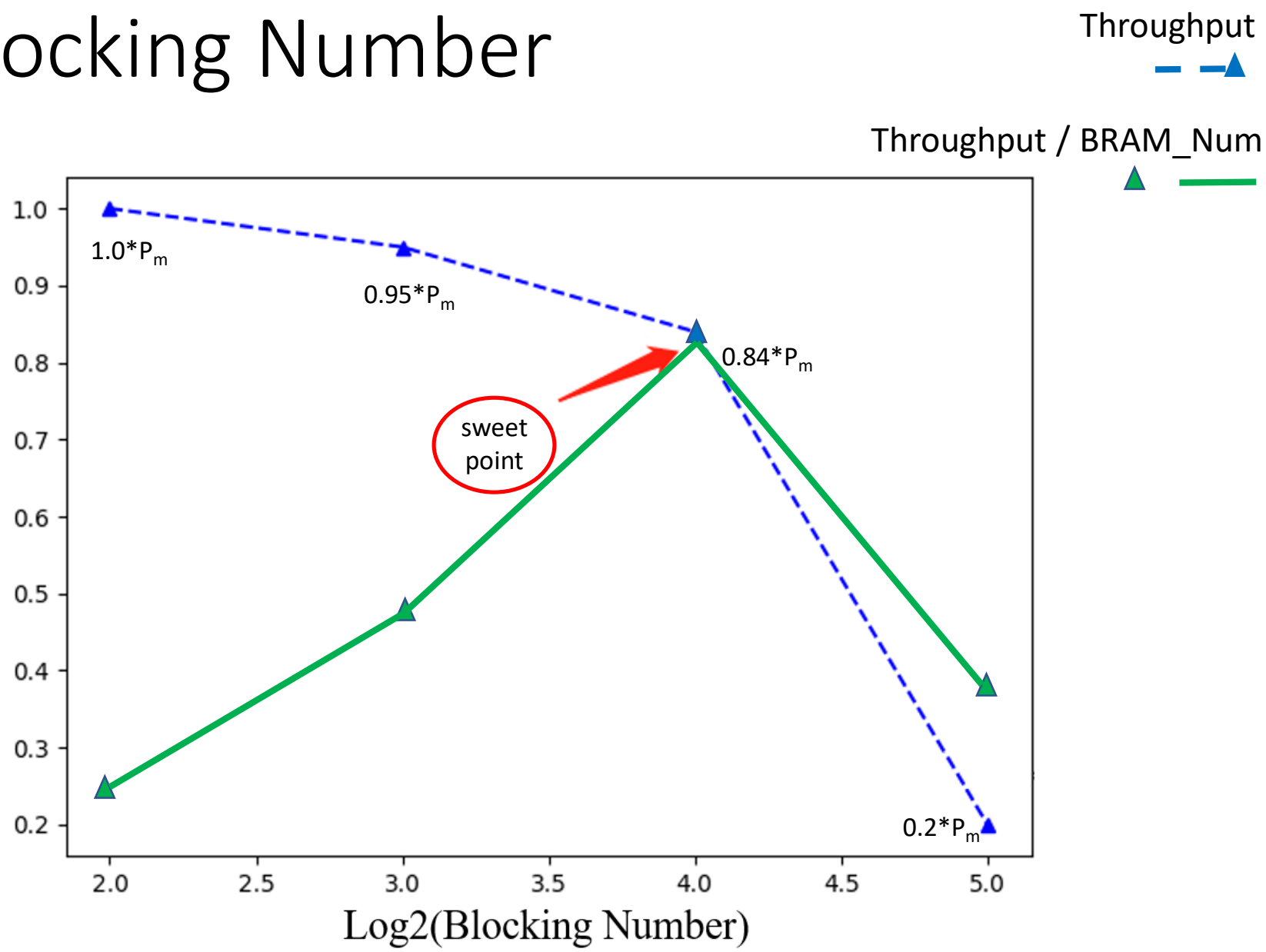
$$P = \frac{M_{op}B}{\frac{M_{op}B}{N_{pe}}} = N_{pe} \quad \text{when } B \geq \frac{N_{pe}}{N_t} \qquad (2)$$

$$P = \frac{M_{op}B}{\frac{M_{op}}{N_t}} = BN_t \quad \text{when } B < \frac{N_{pe}}{N_t} \qquad (3)$$

$$p = \frac{M_{op}B}{\frac{\alpha M_{op}B}{N_{pe}} + \frac{(1-\alpha)M_{op}B}{N_t}} = \frac{N_{pe}N_t}{\alpha N_t + (1-\alpha)N_{pe}}, \; B \geq \frac{N_{pe}}{N_t} \qquad (5)$$

$$p = \frac{M_{op}B}{\frac{\alpha M_{op}}{N_t} + \frac{(1-\alpha)M_{op}B}{N_t}} = \frac{BN_t}{\alpha + (1-\alpha)B}, \; B \leq \frac{N_{pe}}{N_t} \qquad (6)$$

Blocking Number



Outline

- Motivation
- Design & Implementation
  - Stall-free hardware architecture
  - Blocking-batching strategy
  - SBE FPGA Accelerator
- **Evaluation**
  - Batch size and Blocking number
  - **Performance and Efficiency**
- Future Work and Summary

CPU, GPU vs FPGA

Application:
Activity Recognition(LRCN)

CPU&GPU: TensorFlow

	CPU	GPU	This Paper	This Paper
Platform	Intel Xeon E5-2665	TITAN X Pascal	Virtex 7 VX690T	Zynq 7Z045
Frequency	2.4 GHz	1.62 GHz	125 Mhz	142 MHz
Technology	22 nm	16 nm	28 nm	28 nm
Power(W)	93	159	26.5	10.6
Precision	32 bit float		16 bit fixed	
Model Size per Frame[1]	8192[1]* 256			
Time per Sample[2](ms)	14.45	0.78	0.38	0.61
Energy per Sample[2](mJ)	1343	124.02	10.05	6.47

23.7

1.3

208

19.2

[1] Combing the four matrices of i, f, o, g gates.

[2] Each sample/video has 32 frames.

Comparison with other FPGA designs

	2017 [5]	ESE [4]	FP-DNN [13]	This Paper	This Paper
FPGA	Virtex7 VX485T	Kintex KU060	StratixV GSMD5	Virtex7 VX690T	Zynq 7Z045
Model Storage	off-chip				
Prec. (bits)	32[a]	12	16 32[a]	16	16
No. of Used DSP	1176	1504	2072[b]	2060	900
Freq. (Mhz)	150	200	150	125	142
Perf. (GOPS)	7.26	282	316 86[a]	356	221
Power Effi. (GOPS/W)	0.37	6.87	12.63 3.44[a]	13.48	20.84
Resource Effi.[c] (GOPS/DSP)	0.006	0.188	0.153 0.042[a]	0.173	0.246

Fastest

Most Efficient

[a] Floating point

[b] One Intel FPGA DSP includes two 18*18 multipliers

[c] To make a fair comparison, the number of used DSPs is used to calculate GOPS/DSP when evaluating LSTM accelerator

Outline

- Motivation
- Design & Implementation
  - Stall-free hardware architecture
  - Blocking-batching strategy
  - SBE FPGA Accelerator
- Evaluation
  - Batch size and Blocking number
  - Performance and Efficiency
- **Future Work and Summary**

Future Work

- Combing the proposed SBE with pruning methods

- Deploying very large LSTM models using multi-FPGAs

Summary

- Blocking-batching strategy
  - reuse weights for large LSTM systems
- Stall-free architecture
  - reorganise multiplications to enhance throughput
- Zynq and Virtex-7 implementations
  - 23.7x speed, 1/208x energy of CPU
  - 1.3x speed, 1/19.2x energy of GPU