

A11.3—Specification and implementation of individual kernels in hardware

Sažetak—Prema podijeli programa na CPU/FPGA dio, napravljen u prethodnoj fazi projekta, detaljno su specificirane jezgre kojima se ključne petlje implementiraju u FPGA sklopovlju. Specificirane su unaprijeđene verzije jezgri Count conditional, Increment conditional i Logic simple. Unaprijeđenja omogućuju njihovu bolju integraciju u program. Navedene jezgre su implementirane pomoću Maxeler MaxCompiler 2015.2 razvojnog sustava i API-ja, te postavljene da rade na Maxeler Galava kartici.

Specifikacija jezgara

Kao polazna točka uzimaju se jezgre već razvijene u sklopu projekta. Jezgre će se preoblikovati prema slijedećim zahtjevima:

- Jezgre se moraju biti izvedene tako da se mogu međusobno spajati u paralelne i kaskadne kombinacije.
- Implementacija mora biti u obliku *kernel library*, koja omogućuje da se jezgre koriste kao dio drugih, većih, jezgara.
- Jezgre moraju moći primiti podatkovni tok parametara (početne adrese i duljine ulaznih vektora) i na osnovu njih provesti obradu na nizu ulaznih vektora (skupna obrada, *batch processing*).
- Dodjeljivanje vektorskih ulaza i izlaza mora biti što ujednačenije među jezgrama, tj. ista polja u programu bi trebala biti dodijeljena vektorskim ulazima sa istim imenom u svim jezgrama.

Uz to, specifikacija jezgara je preinačena tako da jezgre bolje prate tijek programa.

Count conditional

Skalarni ulazi:

- useCase – odabir načina rada,
- cmpVal (c_{val}) – vrijednost za usporedbu,
- cmpFn (c_{fn}) – odabir funkcije za usporedbu.

Mapa ulaza i izlaza:

	Count coverage	Eliminate covered – A	Eliminate covered – B	Update total coverage	Update OOB – A	Update OOB – B
$sin[0]$	eU			atru	atru	atru
$sin[1]$	selectedp			cls	cls	cls
$sin[2]$	selectedn	selectedn			selected	
$sin[3]$	e					
$count[0]$	m	remCount		kkk	k	k5
$count[1]$	mm			k4	kk	k6

Count enable funkcije:

	Count coverage	Eliminate covered – A	Eliminate covered – B	Update total coverage	Update OOB – A	Update OOB – B
$count[0]$	$\leftarrow s_{in1} (s_{in0} + \overline{s_{in0}} s_{cmp})$	$\leftarrow s_{in2}$		$\leftarrow s_{in0} s_{val}$	$\leftarrow s_{in0} \overline{s_{in2}}$	$\leftarrow s_{in0}$
$count[1]$	$\leftarrow s_{in2} s_{in0} \overline{s_{cmp}}$			$\leftarrow s_{in0}$	$\leftarrow s_{in0} s_{in2} s_{val}$	$\leftarrow s_{in0} s_{val}$

Globalne funkcije:

- $$s_{cmp} \leftarrow \begin{cases} s_{in3} > c_{val}, & \text{za } c_{fn} = 0 \\ s_{in3} \leq c_{val}, & \text{za } c_{fn} = 1 \\ s_{in3} = c_{val}, & \text{za } c_{fn} = 2 \\ s_{in3} \neq c_{val}, & \text{za } c_{fn} = 3 \end{cases}$$
- $s_{val} \leftarrow s_{in1} = c_{val}$

Increment conditional

Skalarni ulazi:

- useCase – odabir načina rada,
- cmpVal (c_{val}) – vrijednost za usporedbu.

Mapa ulaza i izlaza:

	Count coverage	Eliminate covered – A	Eliminate covered – B	Update total coverage	Update OOB – A	Update OOB – B
$sIn[0]$				atru	atru	
$sIn[1]$				cls	selected	
$sIn[2]$				totco	oob	
$sIn[3]$ $sOut[0]$				totco	oob	

Increment enable funkcije:

	Count coverage	Eliminate covered – A	Eliminate covered – B	Update total coverage	Update OOB – A	Update OOB – B
$sOut[0]$				$\leftarrow s_{in0} s_{val}$	$\leftarrow s_{in2}$	

Globalne funkcije:

- $s_{val} \leftarrow s_{in1} = c_{val}$

Logic simple

Skalarni ulazi:

- useCase – odabir načina rada,
- cmpVal (c_{val}) – vrijednost za usporedbu,
- cmpFn (c_{fn}) – odabir funkcije za usporedbu.

Mapa ulaza i izlaza:

	Count coverage	Eliminate covered – A	Eliminate covered – B	Update total coverage	Update OOB – A	Update OOB – B
$sIn[0]$		atru	atru			
$sIn[1]$		selectedp				
$sIn[2]$		selectedn				
$sIn[3]$		e	e			

$sIn[4]$	eU	eU
$sOut[0]$	atru	atru
$sOut[1]$	selectedp	
$sOut[2]$	selectedn	

Logičke funkcije:

	Count coverage	Eliminate covered – A	Eliminate covered – B	Update total coverage	Update OOB – A	Update OOB – B
$sOut[0]$		$\leftarrow s_{in0} \overline{(s_{in4} + \overline{s_{cmp}})}$	$\leftarrow s_{in0} \overline{(s_{in4} + \overline{s_{cmp}})}$			
$sOut[1]$		$\leftarrow s_{in1} \overline{(s_{in4} + \overline{s_{cmp}})}$				
$sOut[2]$		$\leftarrow s_{in2} \overline{(s_{in4} + \overline{s_{cmp}})}$				

Globalne funkcije:

$$\bullet \quad s_{cmp} \leftarrow \begin{cases} s_{in3} > c_{val}, & \text{za } c_{fn} = 0 \\ s_{in3} \leq c_{val}, & \text{za } c_{fn} = 1 \\ s_{in3} = c_{val}, & \text{za } c_{fn} = 2 \\ s_{in3} \neq c_{val}, & \text{za } c_{fn} = 3 \end{cases}$$

Implementacija

Implementacija jezgara je provedena pomoću Maxeler MaxCompiler 2015.2 razvojnog sustava i API-ja. Jezgre su implementirane na Maxeler Galava kartici, s frekvencijom rada 160 MHz.