Many-BSP: An Analytical Performance Model for CUDA Kernels

Supplementary Report

Ali Riahi, Abdorreza Savadi, Mahmoud Naghibzadeh

Computer Engineering Department, Ferdowsi University of Mashhad, Mashhad, Iran

This paper is a supplementary report to the article "Many-BSP: An Analytical Performance Model for CUDA Kernels". The Many-BSP model is introduced in the article and published in the Journal of Parallel and Distributed Computing. Here, details are provided that have been omitted from the article due to the summary. However, we preferred to provide these details to readers in order to describe the model in more detail. It should be noted that this report is not an independent article and along with the main article contains valuable information about the results and how to use the model.

Contents:

1. Introduction

This report is an appendix to the main article entitled " Many-BSP: An Analytical Performance Model for CUDA Kernels ".

Table 1-1, Features of Devices

Video Card	GeForce GTX 760	GeForce 940MX	GeForce GTX 1070
GPU Architecture	Kepler	Maxwell	Pascal
GPU Model	GK104	GM107-B	GP104-A
CUDA Capability (CC)	3.0	5.0	6.1
SM# × Cores/SM	6 × 192	4 × 128	15 × 128
Warp Scheduler#/SM	4	4	4
Dispatch Unit#/Warp Scheduler	2	2	2
Shared Memory Size	16/48 KB	48 KB	48 KB
L1 Cache Size	48/16 KB	48 KB	48 KB
L2 Cache Size	512 KB	1 MB	2 MB
Size of Global Memory	2 GB	2 GB	8 GB
Segment Size of Global Memory	128 B	128 B	128 B
CUDA Version	7.5	9.0	9.0
OS	Ubuntu 16.04	Ubuntu 16.04	Ubuntu 16.04
SPs[1], DPU[2], SFU[3], LDST[4] (per SM)	192, 8, 32, 32	128, 4, 32, 32	128, 4, 32, 32
SPs, DPU, SFU, LDST (Available for one Warp Scheduler)	48, 8, 32, 16	32, 1, 8, 8	32, 1, 8, 8

Table 1-2, Functional units and their instructions throughput for GTX 760 (By ignoring the resource contention)

Functional units	Operations	Throughput	
		Per SM	Per WS[5]
SPs	32-bit floating-point add, multiply, multiply-add	192	32
	32-bit integer add, extended precision add, subtract, extended precision subtract	160	32
	32-bit integer multiply, multiply-add, extended precision multiply-add	32	32
	32-bit integer shift	32	32
	32-bit integer bit reverse, bit field, extract/insert	32	32
	32-bit bitwise AND, OR, XOR	160	32
	compare, minimum, maximum	160	32
	population count	32	32
	sum of absolute difference	32	32
	Type conversions from 8-bit and 16-bit integer to 32-bit types	128	32
	Type conversions from and to 64-bit types	8	8
	All other type conversions	32	32
DPU	64-bit floating-point add, multiply, multiply-add, mov.f64	8	8
SFU	32-bit floating-point SFU (reciprocal, square root, base-2 logarithm, base2 exponential, sine, cosine, …	32	16
LDST	Load and store data to/from memories	32	16

Table 1-3, Functional units and their instructions throughput for 940MX

Functional units	Operations	Throughput	
		Per SM	Per WS
SPs	32-bit integer multiply, multiply-add, extended precision multiply-add	128	32
	32-bit integer add, extended precision add, subtract, extended precision subtract		
	32-bit floating point add, multiply, multiply-add		
	32-bit integer shift	64	16
	32-bit integer bit reverse, bit field, extract/insert	64	16

[1] Scalar Processor (Arithmetic, logical, compare, conversion and control flow), SPs ≡ FU_1
[2] Double Precision Unit (64-bit floating point add, multiply, multiply-add) , DPU ≡ FU_2
[3] Special Function Unit (32-bit floating point SFU) , SFU ≡ FU_3
[4] Load and store data from/to memories , LDST ≡ FU_4
[5] Warp Scheduler

	32-bit bitwise AND, OR, XOR	128	32
	compare, minimum, maximum	64	16
	population count	32	8
	sum of absolute difference	32	8
	Type conversions from 8-bit and 16-bit integer to 32-bit types	32	8
	Type conversions from and to 64-bit types	4	1
	All other type conversions	32	8
DPU	64-bit floating point add, multiply, multiply-add, mov.f64	4	1
SFU	32-bit floating point SFU (reciprocal, square root, base-2 logarithm, base2 exponential, sine, cosine, …	32	8
LDST	Load and store data to/from memories	32	8

Table 1-4, Functional units and their instructions throughput for GTX 1070

Functional units	Operations	Throughput	
		Per SM	Per WS
SPs	32-bit integer multiply, multiply-add, extended precision multiply-add	128	32
	32-bit integer add, extended precision add, subtract, extended precision subtract		
	32-bit floating point add, multiply, multiply-add		
	32-bit integer shift	64	16
	32-bit integer bit reverse, bit field, extract/insert	64	16
	32-bit bitwise AND, OR, XOR	128	32
	compare, minimum, maximum	64	16
	population count	32	8
	sum of absolute difference	64	16
	Type conversions from 8-bit and 16-bit integer to 32-bit types	32	8
	Type conversions from and to 64-bit types	4	1
	All other type conversions	32	8
DPU	64-bit floating point add, multiply, multiply-add, mov.f64	4	1
SFU	32-bit floating point SFU (reciprocal, square root, base-2 logarithm, base2 exponential, sine, cosine, …	32	8
LDST	Load and store data to/from memories	32	8

Table 1-5, Latencies of instructions and their throughputs for GTX 760

FU[1]	Instructions		n_{fu}^{i}	T_i	Latency		Overhead
					Comp.	Comm.	
	bar.sync (for nt=256)		MI	--			173
	bar.sync (for nt=1024)		MI	--			297
SPs	@!%p7 bra	BB0_1;	32	32	16		
	add.f32	%f1, %f2, %f3;	32	32	16		
	add.s32	%r1, %r2, %r3;	32	32	16		
	add.s64	%rd1, %rd2, %rd3;	32	32	16		
	and.b16	%rs1, %rs2, 255;	32	32	16		
	and.b32	%r1, %r2, %r3;	32	32	16		
	and.pred	%p1, %p2, %p3;	32	32	16		
	bra.uni	BB0_1;	32	32	16		
	cvt.f64.f32	%fd1, %f1;	32	8	16		
	cvt.rn.f32.f64	%f23, %fd15;	8	8	16		
	cvta.to.global.u64	%rd1, %rd2;	32	32	16		
	fma.rn.f32	%f10, %f9, %f8, %f103;	32	32	41		
	ld.param.f32	%f1, [Kernel_param_3];	32	32	16		
	ld.param.u32	%r1, [_Kerrn_...];	32	32	16		
	ld.param.u64	%rd1, [Kernel_param_0];	32	32	16		
	mad.lo.s32	%r27, %r1, %r10, %r2;	32	32	16		
	mov.f32	%f103, 0f00000000;	32	32	16		
	mov.u16	%rs1, 0;	32	32	16		
	mov.u32	%r14, %ctaid.x/%tid.x;	32	32	32		
	mov.u32	%r4, %nctaid.x;	32	32	16		
	mov.u32	%r1, %r2;	32	32	16		
	mov.u64	%rd10, Kerr_...;	32	32	16		
	mul.f32	%f22, %f1, %f21;	32	32	16		
	mul.lo/wide.s32	%r19, %r18, %r17;	32	32	16		
	neg.s32	%r1, %r2;	32	32	16		

[1] Functional Unit

	or.pred	%p1, %p2, %p3;	32	32	16		
	setp.gt/ge/lt/le/eq.s32	%p1, %r1, %r2;	32	32	16		
	selp.b32	%r1, %r2, 1, %p1;	32	32	16		
	shl.b32	%r1, %r2, 1;	32	32	16		
	shr.s32	%r1, %r2, 1;	32	32	16		
	sub.f32	%f4, %f1, %f3;	32	32	16		
	sub.s32	%r1, %r2, %r3;	32	32	16		
DPU	add.f64	%fd1, %fd2, %fd3;	8	8	16		
	fma.rn.f64	%fd1, %fd2, %fd3, %fd4;	8	8	46		
	sub.f64	%fd1, %fd2, %fd3;	8	8	16		
SFU	div.rn.f32	%f1, %f2, %f3;	16	16	139		
	rcp.rn.f32	%f1, %f2;	16	16	419		
	sqrt.rn.f32	%f9, %f8;	8	8	411		
LDST	ld.global.f32	%f7, [%rd8];	16	16		191	
	ld.shared.f32	%f1, [%rd1];	16	16	16		
	st.global.f32	[%rd57], %f25;	16	16		191	
	st.shared.f32	[%rd1], %f1;	16	16	41		

Table 1-6, Latencies of instructions and their throughputs for 940MX

FU	Instructions		n_{fu}^{i}	T_i	Latency		Overhead
					Comp.	Comm.	
	bar.sync (for nt=256)		MI	--			120
	bar.sync (for nt=1024)		MI				230
SPs	@!%p7 bra	BB0_2;	32	16	6		
	add.f32	%f16, %f15, %f14;	32	32	6		
	add.s32	%r1, %r20, %r23;	32	32	6		
	add.s64	%rd8, %rd6, %rd7;	32	32	6		
	and.b16	%rs7, %rs8, 255;	32	32	6		
	and.b32	%r79, %r69, %r78;	32	32	6		
	and.pred	%p3, %p1, %p2;	32	32	6		
	bra.uni	BB0_1;	32	32	6		
	cvt.f64.f32	%fd1, %f9;	32	1	6		
	cvt.rn.f32.f64	%f23, %fd15;	32	8	6		
	cvta.to.global.u64	%rd55, %rd5;	32	8	6		
	fma.rn.f32	%f8, %f4, %f4, %f7;	32	32	20		
	ld.param.f32	%f2, [_Kerrnel_param_8];	32	32	6		
	ld.param.u32	%r10, [_Kerrnel_param_4];	32	32	6		
	ld.param.u64	%rd3, [_Kerrnel_param_1];	32	32	6		
	mad.lo.s32	%r27, %r1, %r10, %r2;	32	32	6		
	mov.f32	%f20, 0f42A00000;	32	32	6		
	mov.u16	%rs8, 0;	32	32	6		
	mov.u32	%r16, 16;	32	32	6		
	mov.u32	%r14, %ctaid.x/%tid.x;	32	32	27		
	mov.u32	%r4, %nctaid.x;	32	32	6		
	mov.u64	%rd10, Kerrnel$__cuda_..;	32	32	6		
	mul.f32	%f22, %f1, %f21;	32	32	6		
	mul.lo/wide.s32	%r19, %r18, %r17;	32	32	6		
	neg.s32	%r78, %r22;	32	16	6		
	or.pred	%p18, %p17, %p16;	32	32	6		
	setp.eq.s16	%p34, %rs7, 0;	32	16	6		
	setp.gt/ge/le/lt/eq.s32	%p1, %r1, -1;	32	16	6		
	selp.b32	%r43, %r42, 15, %p9;	32	16	6		
	shl.b32	%r15, %r9, 1;	32	16	6		
	shr.s32	%r69, %r22, 31;	32	16	6		
	sub.f32	%f21, %f20, %f12;	32	32	6		
	sub.s32	%r17, %r16, %r15;	32	32	6		
DPU	add.f64	%fd7, %fd4, %fd4;	1	1	6		
	fma.rn.f64	%fd9, %fd2, %fd8, %fd5;	1	1	65		
	sub.f64	%fd8, %fd6, %fd7;	1	1	6		
SFU	div.rn.f32	%f9, %f6, %f2;	8	8	137		
	rcp.rn.f32	%f10, %f3;	8	8	370		
	sqrt.rn.f32	%f9, %f8;	8	8	370		
LDST	ld.global.f32	%f7, [%rd8];	8	8		313	
	ld.shared.f32	%f12, [%rd30];	8	8	6		
	st.global.f32	[%rd57], %f25;	8	8		313	
	st.shared.f32	[%rd13], %f7;	8	8	20		

Table 1-7, Latencies of instructions and their throughputs for GTX 1070

FU	Instructions	n_{fu}^{i}	T_i	Latency		Overhead
				Comp.	Comm.	
	bar.sync (for nt=256)	MI	--			118
	bar.sync (for nt=1024)	MI				223
SPs	@!%p7 bra BB0_2;	32	16	6		
	add.f32 %f16, %f15, %f14;	32	32	6		
	add.s32 %r1, %r20, %r23;	32	32	6		
	add.s64 %rd8, %rd6, %rd7;	32	32	6		
	and.b16 %rs7, %rs8, 255;	32	32	6		
	and.b32 %r79, %r69, %r78;	32	32	6		
	and.pred %p3, %p1, %p2;	32	32	6		
	bra.uni BB0_1;	32	32	6		
	cvt.f64.f32 %fd1, %f9;	32	1	6		
	cvt.rn.f32.f64 %f23, %fd15;	32	8	6		
	cvta.to.global.u64 %rd55, %rd5;	32	8	6		
	fma.rn.f32 %f8, %f4, %f4, %f7;	32	32	19		
	ld.param.f32 %f2, [_Kerrnel_param_8];	32	32	6		
	ld.param.u32 %r10, [_Kerrnel_param_4];	32	32	6		
	ld.param.u64 %rd3, [_Kerrnel_param_1];	32	32	6		
	mad.lo.s32 %r27, %r1, %r10, %r2;	32	32	6		
	mov.f32 %f20, 0f42A00000;	32	32	6		
	mov.u16 %rs8, 0;	32	32	6		
	mov.u32 %r16, 16;	32	32	6		
	mov.u32 %r14, %ctaid.x/%tid.x;	32	32	29		
	mov.u32 %r4, %nctaid.x;	32	32	6		
	mov.u64 %rd10, Kerrnel$__cuda_..;	32	32	6		
	mul.f32 %f22, %f1, %f21;	32	32	6		
	mul.lo/wide.s32 %r19, %r18, %r17;	32	32	6		
	neg.s32 %r78, %r22;	32	16	6		
	or.pred %p18, %p17, %p16;	32	32	6		
	setp.eq.s16 %p34, %rs7, 0;	32	16	6		
	setp.gt/ge/le/lt/eq.s32 %p1, %r1, -1;	32	16	6		
	selp.b32 %r43, %r42, 15, %p9;	32	16	6		
	shl.b32 %r15, %r9, 1;	32	16	6		
	shr.s32 %r69, %r22, 31;	32	16	6		
	sub.f32 %f21, %f20, %f12;	32	32	6		
	sub.s32 %r17, %r16, %r15;	32	32	6		
DPU	add.f64 %fd7, %fd4, %fd4;	1	1	6		
	fma.rn.f64 %fd9, %fd2, %fd8, %fd5;	1	1	63		
	sub.f64 %fd8, %fd6, %fd7;	1	1	6		
SFU	div.rn.f32 %f9, %f6, %f2;	8	8	133		
	rcp.rn.f32 %f10, %f3;	8	8	366		
	sqrt.rn.f32 %f9, %f8;	8	8	366		
LDST	ld.global.f32 %f7, [%rd8];	8	8		394	
	ld.shared.f32 %f12, [%rd30];	8	8	6		
	st.global.f32 [%rd57], %f25;	8	8		394	
	st.shared.f32 [%rd13], %f7;	8	8	20		

The following steps should be followed to perform experiments and calculate the execution time of a CUDA kernel through Many-BSP model:

Step 1: First, the initial parameters listed in Table 1-8 are set.

Table 1-8, The initial parameters in Many-BSP model

d	n_b	g_0	max_*thread_per_sm*
n_c	n_t	g_1	*reg_per_thread*
n_{SM}	h	g_2	*shmem_size*
n_{ws}	l	$mem\_lat = G_2$	*shmem_per_block*
n_{du}	l_c	*warp_lnch_ovh*	n_{reg}
n_{fu}	l_m	*block_lnch_ovh*	*warp_size*
w	l_b	*issue_cycle*	

Step 2: Create Table 1-9 for an application, based on its PTX code.

Table 1-9, PTX code of application and analysis of instruction latencies and vectors u, d, and, p

Inst#	Instructions	FU	n_{fu}^{i}	T_i	n_{ma}^{i}	Latency		u	d	p
						Comp.	Comm.			

Step 3: Create a BLOCK data structure similar to Table 1-10. Characteristics such as multi-threading, branch divergence, ILP, TLP, coalesced access to global memory, shared memory bank conflict, parallel execution of instructions in functional units, and computational instruction pipelines are considered at this step. In This step, the number of accesses to the memory hierarchy (n_{ma}^{i}) is estimated by simultaneously analyzing the CUDA and PTX codes. Also, for each instruction, the functional unit that executes it is specified.

Table 1-10, BLOCK data structure

Inst#	Instructions	$comp_i$						$comm_i$	ovh_i	$sync_i$	u	d	p
		Issue	FU_1	FU_2	FU_3	FU_4	$warp\_size/T_i$						

Step 4: Using the BLOCK data structure, level 1 and 2 supersteps are identified and their costs are estimated. Also, the number of iterations of level one supersteps is determined.

Step 5: Using equations 10 to 25, the kernel execution time is estimated.

Step 6: The kernel execution time is measured by profiling and the prediction error of the model is calculated by the following equation:

$$error\ \% = \frac{|mesured\_exec\_cycle - kernel\_exec\_cycle|}{mesured\_exec\_cycle} \times 100$$

The equations used in the model are as follows:

$$warps\_need = \frac{mem\_lat}{inst\_time \times \alpha} \tag{1}$$

$$warps\_need = \frac{mem\_lat}{inst\_time \times (\alpha + 1)} + 1 \tag{2}$$

$$warps\_need = mem\_lat \times \frac{alu\_thru}{\alpha} + alu\_lat \times alu\_thru \tag{3}$$

$$block\_exec\_cycle = block\_lnch\_ovh + block\_bar\_ovh + nonoverlapped + COMP \tag{4}$$

$$COMP = w \times warp\_lnch\_ovh + parallel\_comp \tag{5}$$

$$w = \left\lceil \frac{n_t}{warp\_size \times n_{ws}} \right\rceil \tag{6}$$

$$\sum_{k=l}^{i} block[k].FU[x] \leq \sum_{k=l+1}^{j-1} issue\_cycle_k + \max_{y=1\ldots n_{fu}, y\neq x} \sum_{k=l}^{j-2} block[k].FU[y] \tag{7}$$

$$block[i].sync = \sum_{k=l}^{i} block[k].FU[x] - \left(\sum_{k=l+1}^{j-1} issue\_cycle_k + \max_{y=1\ldots n_{fu}, y\neq x} \sum_{k=l}^{j-2} block[k].FU[y] \right) \tag{8}$$

$$comp_t = block[i].issue\_cycle + \max_{q=1\ldots n_{fu}} \left\{ \sum_{k=i+1}^{j} block[k].issue\_cycle\,, \sum_{k=i}^{j} block[k].FU[q] \right\} \tag{9}$$

$$parallel\_comp = \sum_{i=1}^{s_1} t_i \times sstp1[i].comp \tag{10}$$

$$block\_bar\_ovh = \sum_{i=1}^{s_1} t_i \times sstp1[i].ovh \tag{11}$$

$$block\_comm = \sum_{i=1}^{s_1} t_i \times sstp1[i].comm \tag{12}$$

$$\sum_{i=0}^{w-1} overlapped_i \le max\_sum\_overlapped = \min\left\{ block\_comm\,, \frac{w(w-1)}{2} + (COMP - w) \times (w-1) \right\} \tag{13}$$

$$min\_nonoverlapped = \left\lfloor \frac{block\_comm - max\_sum\_overlapped}{w} \right\rfloor \tag{14}$$

$$warps\_need = n_{ws} \times \left(\left\lceil \frac{\frac{warp\_comm_\Delta}{l_m - 1}}{\frac{warp\_comp}{l_c}} \right\rceil + 1 \right) = n_{ws} \times \left(\left\lceil \frac{warp\_comm_\Delta \times l_c}{warp\_comp \times (l_m - 1)} \right\rceil + 1 \right) \tag{15}$$

$$overlapped = \min\left(1, \frac{w \times n_{ws}}{warps\_need}\right) \times \frac{block\_comm_\Delta}{w} \tag{16}$$

$$\begin{aligned} nonoverlapped &= \min\left\{ \frac{block\_comm}{w}, mem\_lat + \frac{block\_comm_\Delta}{w} - overlapped \right\} \\ &= \min\left\{ \frac{block\_comm}{w}, mem\_lat + \frac{block\_comm_\Delta}{w} - \min\left(1, \frac{w \times n_{ws}}{warps\_need}\right) \times \frac{block\_comm_\Delta}{w} \right\} \\ &= \min\left\{ \frac{block\_comm}{w}, mem\_lat + \frac{block\_comm_\Delta}{w} \times \max\left(0, 1 - \frac{w \times n_{ws}}{warps\_need}\right) \right\} \end{aligned} \tag{17}$$

$$block\_profile_i =< COMP_i, nonoverlapped_i, block\_bar\_ovh_i >, \quad i = 0 \dots n_b - 1 \tag{18}$$

$$comp = \frac{1}{n_b} \times \sum_{i=0}^{n_b - 1} (COMP_i + block\_bar\_ovh_i) \tag{19}$$

$$novlp = \frac{1}{n_b} \times \sum_{i=0}^{n_b - 1} nonoverlapped_i \tag{20}$$

$$\rho = \max\{ 1, \min\{ \left\lfloor \frac{\max\_thread\_per\_sm}{n_t} \right\rfloor, \left\lfloor \frac{n_{reg}}{n_t \times reg\_per\_thread} \right\rfloor, \left\lfloor \frac{shmem\_size}{shmem\_per\_block} \right\rfloor \} \} \tag{21}$$

$$\mathrm{K} = \frac{n_b}{n_{sm} \times \rho} \tag{22}$$

$$\tau = \left\lceil \frac{novlp}{comp} \right\rceil + 1 \tag{23}$$

$$if\ \rho \ge \tau\ then\ kernel\_exec\_cycle = block\_lnch\_ovh + \frac{n_b}{n_{sm}} \times \frac{comp}{\min\{\mu, \frac{1+\rho}{2}\}} + \frac{novlp}{2} \tag{24}$$

$$\begin{aligned} if\ \rho < \tau\ then\ kernel\_exec\_cycle \\ = block\_lnch\_ovh + \frac{n_b}{n_{sm}} \times \frac{comp}{\min\{\mu, \frac{1+\rho}{2}\}} + (\mathrm{K} - 1) \times \frac{\tau - \rho}{\tau - 1} \times novlp + \frac{novlp}{2} \end{aligned} \tag{25}$$

$$G_i = g_0 + g_1 + \cdots + g_i \tag{26}$$

$$error\ \% = \frac{|mesured\_exec\_cycle - kernel\_exec\_cycle|}{mesured\_exec\_cycle} \times 100 \tag{27}$$

2. Details of results for Hotspot on GTX 760

$d = 3$	$n_b = 1849$	$g_0 = 32$	$\max\_thread\_per\_sm = 2048$
$n_c = 192$	$n_t = 256$	$g_1 = 98$	$reg\_per\_thread = 34$
$n_{SM} = 6$	$h = 196$	$g_2 = 61$	$shmem\_size = 48\ KB$
$n_{ws} = 4$	$l = 272$	$mem\_lat = G_2 = 191$	$shmem\_per\_block = 3072\ B$
$n_{du} = 8$	$l_c = 265$	$warp\_lnch\_ovh = 10$	$n_{reg} = 65536$
$n_{fu} = 4$	$l_m = 3$	$block\_lnch\_ovh = 553$	
$warp\_size = 32$	$l_b = 4$	$issue\_cycle\ = 1$	

$$w\ (eq.\ 6) = \left\lceil \frac{256}{32 \times 4} \right\rceil = 2$$

Table 2-1, Latency, FUs, Throughput, u, d and p for Hotspot on GTX 760

Inst#	Instructions	FU	n^i_{fu}	T_i	n^i_{ma}	Latency		u	d	p
						Comp.	Comm.			
1	ld.param.u32 %r9, [_Kerrnel_param_0];	SPs	32	32		16		15	0	0
2	ld.param.u64 %rd3, [_Kerrnel_param_1];	SPs	32	32		16		49	0	0
3	ld.param.u64 %rd4, [_Kerrnel_param_2];	SPs	32	32		16		39	0	0
4	ld.param.u64 %rd5, [_Kerrnel_param_3];	SPs	32	32		16		193	0	0
5	ld.param.u32 %r10, [_Kerrnel_param_4];	SPs	32	32		16		32	0	0
6	ld.param.u32 %r11, [_Kerrnel_param_5];	SPs	32	32		16		28	0	0
7	ld.param.u32 %r12, [_Kerrnel_param_6];	SPs	32	32		16		22	0	0
8	ld.param.u32 %r13, [_Kerrnel_param_7];	SPs	32	32		16		20	0	0
9	ld.param.f32 %f2, [_Kerrnel_param_8];	SPs	32	32		16		59	0	0
10	ld.param.f32 %f3, [_Kerrnel_param_9];	SPs	32	32		16		60	0	0
11	ld.param.f32 %f4, [_Kerrnel_param_10];	SPs	32	32		16		61	0	0
12	ld.param.f32 %f5, [_Kerrnel_param_11];	SPs	32	32		16		62	0	0
13	ld.param.f32 %f6, [_Kerrnel_param_12];	SPs	32	32		16		59	0	0
14	mov.u32 %r14, %ctaid.x;	SPs	32	32		32		21	0	0
15	shl.b32 %r15, %r9, 1;	SPs	32	32		16		17	0	0
16	mov.u32 %r16, 16;	SPs	32	32		16		17	1	0
17	sub.s32 %r17, %r16, %r15;	SPs	32	32		16		19	0	0
18	mov.u32 %r18, %ctaid.y;	SPs	32	32		32		19	1	0
19	mul.lo.s32 %r19, %r18, %r17;	SPs	32	32		16		20	1	0
20	sub.s32 %r20, %r19, %r13;	SPs	32	32		16		24	0	0
21	mul.lo.s32 %r21, %r14, %r17;	SPs	32	32		16		22	1	0
22	sub.s32 %r22, %r21, %r12;	SPs	32	32		16		26	0	0
23	mov.u32 %r23, %tid.y;	SPs	32	32		32		24	1	0
24	add.s32 %r1, %r20, %r23;	SPs	32	32		16		27	0	0
25	mov.u32 %r24, %tid.x;	SPs	32	32		32		26	1	0
26	add.s32 %r2, %r22, %r24;	SPs	32	32		16		33	0	0
27	setp.gt.s32 %p1, %r1, -1;	SPs	32	32		16		30	0	0
28	add.s32 %r25, %r11, -1;	SPs	32	32		16		29	1	0
29	setp.le.s32 %p2, %r1, %r25;	SPs	32	32		16		30	1	0
30	and.pred %p3, %p1, %p2;	SPs	32	32		16		35	0	0
31	setp.gt.s32 %p4, %r2, -1;	SPs	32	32		16		34	0	0
32	add.s32 %r26, %r10, -1;	SPs	32	32		16		33	1	0
33	setp.le.s32 %p5, %r2, %r26;	SPs	32	32		16		34	1	0
34	and.pred %p6, %p4, %p5;	SPs	32	32		16		35	1	0
35	and.pred %p7, %p3, %p6;	SPs	32	32		16		36	1	0
36	@!%p7 bra BB0_2;	SPs	32	32		16		0	0	0
37	bra.uni BB0_1;	SPs	32	32		16		0	0	0
--	BB0_1:							--	--	--
38	mad.lo.s32 %r27, %r1, %r10, %r2;	SPs	32	32		16		40	0	0
39	cvta.to.global.u64 %rd6, %rd4;	SPs	32	32		16		41	0	0
40	mul.wide.s32 %rd7, %r27, 4;	SPs	32	32		16		41	1	0
41	add.s64 %rd8, %rd6, %rd7;	SPs	32	32		16		42	1	1
42	ld.global.f32 %f7, [%rd8];	LDST	16	16	1~2		191	48	0	1
43	mul.wide.s32 %rd9, %r23, 64;	SPs	32	32		16		45	0	0
44	mov.u64 %rd10, Kerrnel$__cuda_...;	SPs	32	32		16		45	1	0
45	add.s64 %rd11, %rd10, %rd9;	SPs	32	32		16		47	0	0
46	mul.wide.s32 %rd12, %r24, 4;	SPs	32	32		16		47	1	0
47	add.s64 %rd13, %rd11, %rd12;	SPs	32	32		16		48	1	1
48	st.shared.f32 [%rd13], %f7;	LDST	16	16	1	41		0	0	1
49	cvta.to.global.u64 %rd14, %rd3;	SPs	32	32		16		50	1	0
50	add.s64 %rd15, %rd14, %rd7;	SPs	32	32		16		51	1	1
51	ld.global.f32 %f8, [%rd15];	LDST	16	16	1~2		191	55	0	1
52	mov.u64 %rd16, _Kerrnel$__cuda_...;	SPs	32	32		16		53	1	0
53	add.s64 %rd17, %rd16, %rd9;	SPs	32	32		16		54	1	0
54	add.s64 %rd18, %rd17, %rd12;	SPs	32	32		16		55	1	1

55	st.shared.f32 [%rd18], %f8;	LDST	16	16	1	41		0	0	0
--	BB0_2:							--	--	--
56	bar.sync 0;		MI[1]	--		173		0	0	0
57	setp.lt.s32 %p8, %r9, 1;	SPs	32	32		16		58	1	0
58	@%p8 bra BB0_11;	SPs	32	32		16		0	0	0
59	div.rn.f32 %f9, %f6, %f2;	SFU	16	16		139		63	0	0
60	rcp.rn.f32 %f10, %f3;	SFU	16	16		419		93	0	0
61	rcp.rn.f32 %f11, %f4;	SFU	16	16		419		79	0	0
62	rcp.rn.f32 %f1, %f5;	SFU	16	16		419		158	0	1
63	cvt.f64.f32 %fd1, %f9;	SPs	32	8		16		161	0	0
64	add.s32 %r37, %r20, 15;	SPs	32	32		16		65	1	0
65	setp.gt.s32 %p9, %r37, %r25;	SPs	32	32		16		70	0	0
66	mov.u32 %r39, -15;	SPs	32	32		16		67	1	0
67	sub.s32 %r40, %r39, %r20;	SPs	32	32		16		68	1	0
68	add.s32 %r41, %r11, %r40;	SPs	32	32		16		69	1	0
69	add.s32 %r42, %r41, 14;	SPs	32	32		16		70	1	0
70	selp.b32%r43, %r42, 15, %p9;	SPs	32	32		16		71	1	0
71	setp.lt.s32 %p10, %r23, %r43;	SPs	32	32		16		73	0	0
72	add.s32 %r45, %r23, 1;	SPs	32	32		16		73	1	0
73	selp.b32%r46, %r45, %r43, %p10;	SPs	32	32		16		74	1	0
74	mul.wide.s32 %rd19, %r46, 64;	SPs	32	32		16		76	0	0
75	mov.u64 %rd20, _Kerrnel$__cuda_...;	SPs	32	32		16		76	1	0
76	add.s64 %rd21, %rd20, %rd19;	SPs	32	32		16		78	0	0
77	mul.wide.s32 %rd22, %r24, 4;	SPs	32	32		16		78	1	0
78	add.s64 %rd1, %rd21, %rd22;	SPs	32	32		16		139	0	0
79	cvt.f64.f32 %fd2, %f11;	SPs	32	8		16		144	0	0
80	add.s32 %r51, %r22, 15;	SPs	32	32		16		81	1	0
81	setp.gt.s32 %p11, %r51, %r26;	SPs	32	32		16		85	0	0
82	sub.s32 %r53, %r39, %r22;	SPs	32	32		16		83	1	0
83	add.s32 %r54, %r10, %r53;	SPs	32	32		16		84	1	0
84	add.s32 %r55, %r54, 14;	SPs	32	32		16		85	1	0
85	selp.b32%r56, %r55, 15, %p11;	SPs	32	32		16		86	1	0
86	setp.lt.s32 %p12, %r24, %r56;	SPs	32	32		16		88	0	0
87	add.s32 %r57, %r24, 1;	SPs	32	32		16		88	1	0
88	selp.b32%r58, %r57, %r56, %p12;	SPs	32	32		16		89	1	0
89	mul.wide.s32 %rd23, %r58, 4;	SPs	32	32		16		92	0	0
90	mul.wide.s32 %rd24, %r23, 64;	SPs	32	32		16		91	1	0
91	add.s64 %rd25, %rd20, %rd24;	SPs	32	32		16		92	1	0
92	add.s64 %rd2, %rd25, %rd23;	SPs	32	32		16		151	0	0
93	cvt.f64.f32 %fd3, %f10;	SPs	32	8		16		155	0	0
94	mov.u32 %r59, 1;	SPs	32	32		16		95	1	0
95	sub.s32 %r129, %r59, %r9;	SPs	32	32		16		169	0	0
96	mov.u32 %r130, 0;	SPs	32	32		16		98	0	0
--	BB0_4:							--	--	--
97	mov.u32 %r61, 14;	SPs	32	32		16		98	1	0
98	sub.s32 %r6, %r61, %r130;	SPs	32	32		16		99	1	0
99	setp.le.s32 %p13, %r24, %r6;	SPs	32	32		16		102	0	0
100	add.s32 %r130, %r130, 1;	SPs	32	32		16		101	1	0
101	setp.ge.s32 %p14, %r24, %r130;	SPs	32	32		16		102	1	0
102	and.pred %p15, %p13, %p14;	SPs	32	32		16		104	0	0
103	mov.u16 %rs8, 0;	SPs	32	32		16		171	0	0
104	@!%p15 bra BB0_7;	SPs	32	32		16		0	0	0
105	bra.uni BB0_5;	SPs	32	32		16		0	0	0
--	BB0_5:							--	--	--
106	setp.gt.s32 %p16, %r23, %r6;	SPs	32	32		16		108	0	0
107	setp.lt.s32 %p17, %r23, %r130;	SPs	32	32		16		108	1	0
108	or.pred %p18, %p17, %p16;	SPs	32	32		16		115	0	0
109	shr.s32 %r69, %r22, 31;	SPs	32	32		16		112	0	0
110	setp.gt.s32 %p20, %r24, %r56;	SPs	32	32		16		114	0	0
111	neg.s32 %r78, %r22;	SPs	32	32		16		112	1	0
112	and.b32 %r79, %r69, %r78;	SPs	32	32		16		113	1	0
113	setp.lt.s32 %p21, %r24, %r79;	SPs	32	32		16		114	1	0
114	or.pred %p22, %p21, %p20;	SPs	32	32		16		115	1	0
115	or.pred %p23, %p18, %p22;	SPs	32	32		16		122	0	0
116	shr.s32 %r83, %r20, 31;	SPs	32	32		16		119	0	0
117	setp.gt.s32 %p25, %r23, %r43;	SPs	32	32		16		121	0	0
118	neg.s32 %r90, %r20;	SPs	32	32		16		119	1	0
119	and.b32 %r91, %r83, %r90;	SPs	32	32		16		120	1	0
120	setp.lt.s32 %p26, %r23, %r91;	SPs	32	32		16		121	1	0
121	or.pred %p27, %p26, %p25;	SPs	32	32		16		122	1	0
122	or.pred %p28, %p23, %p27;	SPs	32	32		16		123	1	0
123	@%p28 bra BB0_7;	SPs	32	32		16		0	0	0

[1] Multiple Instructions

124	add.s64 %rd30, %rd25, %rd22;	SPs	32	32		16		125	1	1
125	ld.shared.f32 %f12, [%rd30];	LDST	16	16	1	16		126	1	1
126	cvt.f64.f32 %fd4, %f12;	SPs	32	8		16		142	0	0
127	mov.u64 %rd31, Kerrnel$__cuda_...;	SPs	32	32		16		128	1	0
128	add.s64 %rd32, %rd31, %rd24;	SPs	32	32		16		129	1	0
129	add.s64 %rd33, %rd32, %rd22;	SPs	32	32		16		130	1	1
130	ld.shared.f32 %f13, [%rd33];	LDST	16	16	1	16		131	1	1
131	cvt.f64.f32 %fd5, %f13;	SPs	32	8		16		144	0	0
132	setp.gt.s32 %p29, %r23, %r91;	SPs	32	32		16		134	0	0
133	add.s32 %r103, %r23, -1;	SPs	32	32		16		134	1	0
134	selp.b32 %r104, %r103, %r91, %p29;	SPs	32	32		16		135	1	0
135	mul.wide.s32 %rd34, %r104, 64;	SPs	32	32		16		136	1	0
136	add.s64 %rd35, %rd20, %rd34;	SPs	32	32		16		137	1	0
137	add.s64 %rd36, %rd35, %rd22;	SPs	32	32		16		138	1	1
138	ld.shared.f32 %f14, [%rd36];	LDST	16	16	1	16		140	0	0
139	ld.shared.f32 %f15, [%rd1];	LDST	16	16	1	16		140	1	1
140	add.f32 %f16, %f15, %f14;	SPs	32	32		16		141	1	0
141	cvt.f64.f32 %fd6, %f16;	SPs	32	8		16		143	0	1
142	add.f64 %fd7, %fd4, %fd4;	DPU	8	8		16		143	1	0
143	sub.f64 %fd8, %fd6, %fd7;	DPU	8	8		16		144	1	0
144	fma.rn.f64 %fd9, %fd2, %fd8, %fd5;	DPU	8	8		46		155	0	1
145	setp.gt.s32 %p30, %r24, %r79;	SPs	32	32		16		147	0	0
146	add.s32 %r111, %r24, -1;	SPs	32	32		16		147	1	0
147	selp.b32 %r112, %r111, %r79, %p30;	SPs	32	32		16		148	1	0
148	mul.wide.s32 %rd37, %r112, 4;	SPs	32	32		16		149	1	0
149	add.s64 %rd38, %rd25, %rd37;	SPs	32	32		16		150	1	1
150	ld.shared.f32 %f17, [%rd38];	LDST	16	16	1	16		152	0	0
151	ld.shared.f32 %f18, [%rd2];	LDST	16	16	1	16		152	1	1
152	add.f32 %f19, %f18, %f17;	SPs	32	32		16		153	1	0
153	cvt.f64.f32 %fd10, %f19;	SPs	32	8		16		154	1	0
154	sub.f64 %fd11, %fd10, %fd7;	DPU	8	8		16		155	1	0
155	fma.rn.f64 %fd12, %fd3, %fd11, %fd9;	DPU	8	8		46		160	0	1
156	mov.f32 %f20, 0f42A00000;	SPs	32	32		16		157	1	0
157	sub.f32 %f21, %f20, %f12;	SPs	32	32		16		158	1	0
158	mul.f32 %f22, %f1, %f21;	SPs	32	32		16		159	1	0
159	cvt.f64.f32 %fd13, %f22;	SPs	32	8		16		160	1	0
160	add.f64 %fd14, %fd13, %fd12;	DPU	8	8		16		161	1	0
161	fma.rn.f64 %fd15, %fd1, %fd14, %fd4;	DPU	8	8		46		162	1	1
162	cvt.rn.f32.f64 %f23, %fd15;	SPs	8	8		16		166	0	0
163	mov.u64 %rd39, _Kerrnel$__cuda_...;	SPs	32	32		16		164	1	0
164	add.s64 %rd40, %rd39, %rd24;	SPs	32	32		16		165	1	0
165	add.s64 %rd41, %rd40, %rd22;	SPs	32	32		16		166	1	1
166	st.shared.f32 [%rd41], %f23;	LDST	16	16	1	41		0	0	1
167	mov.u16 %rs8, 1;	SPs	32	32		16		171	0	0
--	BB0_7:							--	--	--
168	bar.sync 0;		MI			173		0	0	0
169	setp.eq.s32 %p31, %r129, 0;	SPs	32	32		16		170	1	0
170	@%p31 bra BB0_11;	SPs	32	32		16		0	0	0
171	setp.eq.s16 %p32, %rs8, 0;	SPs	32	32		16		172	1	0
172	@%p32 bra BB0_10;	SPs	32	32		16		0	0	0
173	mov.u64 %rd43, _Kerrnel$__cuda_...;	SPs	32	32		16		174	1	0
174	add.s64 %rd44, %rd43, %rd24;	SPs	32	32		16		175	1	0
175	add.s64 %rd46, %rd44, %rd22;	SPs	32	32		16		176	1	1
176	ld.shared.f32 %f24, [%rd46];	LDST	16	16	1	16		178	0	1
177	add.s64 %rd49, %rd25, %rd22;	SPs	32	32		16		178	1	1
178	st.shared.f32 [%rd49], %f24;	LDST	16	16	1	41		0	0	0
--	BB0_10:							--	--	--
179	bar.sync 0;		MI			173		0	0	0
180	add.s32 %r129, %r129, 1;	SPs	32	32		16		0	0	0
181	setp.lt.s32 %p33, %r130, %r9;	SPs	32	32		16		182	1	0
182	@%p33 bra BB0_4;	SPs	32	32		16		0	0	0
--	BB0_11:							--	--	--
183	and.b16 %rs7, %rs8, 255;	SPs	32	32		16		184	1	0
184	setp.eq.s16 %p34, %rs7, 0;	SPs	32	32		16		185	1	0
185	@%p34 bra BB0_13;	SPs	32	32		16		0	0	0
186	mul.wide.s32 %rd50, %r23, 64;	SPs	32	32		16		188	0	0
187	mov.u64 %rd51, _Kerrnel$__cuda_...;	SPs	32	32		16		188	1	0
188	add.s64 %rd52, %rd51, %rd50;	SPs	32	32		16		190	0	0
189	mul.wide.s32 %rd53, %r24, 4;	SPs	32	32		16		190	1	0
190	add.s64 %rd54, %rd52, %rd53;	SPs	32	32		16		191	1	1
191	ld.shared.f32 %f25, [%rd54];	LDST	16	16	1	16		196	0	1
192	mad.lo.s32 %r128, %r1, %r10, %r2;	SPs	32	32		16		194	0	0
193	cvta.to.global.u64 %rd55, %rd5;	SPs	32	32		16		195	0	0
194	mul.wide.s32 %rd56, %r128, 4;	SPs	32	32		16		195	1	0

195	add.s64 %rd57, %rd55, %rd56;	SPs	32	32		16		196	1	1
196	st.global.f32 [%rd57], %f25;	LDST	16	16	1~2		191	0	0	0
--	BB0_13:									
--	ret;									

Table 2-2, BLOCK data structure for Hotspot on GTX 760

Inst#	Instructions	$comp_i$						$comm_i$	ovh_i	$sync_i$	u	d	p
		Issue	FU_1	FU_2	FU_3	FU_4	$32^1/T_i$						
1	ld.param.u32	2	16+1				1			0	15	0	0
2	ld.param.u64	2	2				1			0	49	0	0
3	ld.param.u64	2	2				1			0	39	0	0
4	ld.param.u64	2	2				1			0	193	0	0
5	ld.param.u32	2	2				1			0	32	0	0
6	ld.param.u32	2	2				1			0	28	0	0
7	ld.param.u32	2	2				1			0	22	0	0
8	ld.param.u32	2	2				1			0	20	0	0
9	ld.param.f32	2	2				1			0	59	0	0
10	ld.param.f32	2	2				1			0	60	0	0
11	ld.param.f32	2	2				1			0	61	0	0
12	ld.param.f32	2	2				1			0	62	0	0
13	ld.param.f32	2	2				1			0	59	0	0
14	mov.u32	2	2				1			0	21	0	0
15	shl.b32	2	2				1			0	17	0	0
16	mov.u32	2	2				1			2	17	1	0
17	sub.s32	2	16+1				1			0	19	0	0
18	mov.u32	2	2				1			2	19	1	0
19	mul.lo.s32	2	16+1				1			17	20	1	0
20	sub.s32	2	16+1				1			0	24	0	0
21	mul.lo.s32	2	2				1			2	22	1	0
22	sub.s32	2	16+1				1			0	26	0	0
23	mov.u32	2	2				1			2	24	1	0
24	add.s32	2	16+1				1			0	27	0	0
25	mov.u32	2	2				1			2	26	1	0
26	add.s32	2	16+1				1			0	33	0	0
27	setp.gt.s32	2	2				1			0	30	0	0
28	add.s32	2	2				1			2	29	1	0
29	setp.le.s32	2	16+1				1			17	30	1	0
30	and.pred	2	16+1				1			0	35	0	0
31	setp.gt.s32	2	2				1			0	34	0	0
32	add.s32	2	2				1			2	33	1	0
33	setp.le.s32	2	16+1				1			17	34	1	0
34	and.pred	2	16+1				1			17	35	1	0
35	and.pred	2	16+1				1			17	36	1	0
36	@!%p7 bra BB0_2;	2	16+1				1			1	0	0	0
37	bra.uni BB0_1;	2	2				1			0	0	0	0
--	BB0_1:										--	--	--
38	mad.lo.s32	2	16+1				1			15	40	0	0
39	cvta.to.global.u64	2	2				1			0	41	0	0
40	mul.wide.s32	2	2				1			2	41	1	0
41	add.s64	2	16+1				1			17	42	1	1
42	ld.global.f32	4					2	764		0	48	0	1
43	mul.wide.s32		2				1			1	45	0	0
44	mov.u64	2	2				1			2	45	1	0
45	add.s64	2	16+1				1			0	47	0	0
46	mul.wide.s32	2	2				1			2	47	1	0
47	add.s64	2	16+1				1			17	48	1	1
48	st.shared.f32	4				41+2	2			0	0	0	1
49	cvta.to.global.u64		2				1			2	50	1	0
50	add.s64	2	16+1				1			17	51	1	1
51	ld.global.f32	4					2	764		0	55	0	1
52	mov.u64		2				1			1	53	1	0
53	add.s64	2	16+1				1			17	54	1	0
54	add.s64	2	16+1				1			17	55	1	1
55	st.shared.f32	4				41+2	2			0	0	0	0
--	BB0_2:										--	--	--
56	bar.sync	2					--		173	1	0	0	0
57	setp.lt.s32	2	16+1				1			17	58	1	0
58	@%p8 bra BB0_11;	2	16+1				1			1	0	0	0
59	div.rn.f32	4			139+2		2			129	63	0	0
60	rcp.rn.f32	4			419+2		2			0	93	0	0
61	rcp.rn.f32	4			4		2			0	79	0	0

[1] warp_size

62	rcp.rn.f32	4			4		2			0	158	0	1
63	cvt.f64.f32		8				4			0	161	0	0
64	add.s32	2	2				1			2	65	1	0
65	setp.gt.s32	2	16+1				1			0	70	0	0
66	mov.u32	2	2				1			2	67	1	0
67	sub.s32	2	16+1				1			17	68	1	0
68	add.s32	2	16+1				1			17	69	1	0
69	add.s32	2	16+1				1			17	70	1	0
70	selp.b32	2	16+1				1			17	71	1	0
71	setp.lt.s32	2	16+1				1			0	73	0	0
72	add.s32	2	2				1			2	73	1	0
73	selp.b32	2	16+1				1			17	74	1	0
74	mul.wide.s32	2	16+1				1			0	76	0	0
75	mov.u64	2	2				1			2	76	1	0
76	add.s64	2	16+1				1			0	78	0	0
77	mul.wide.s32	2	2				1			2	78	1	0
78	add.s64	2	16+1				1			0	139	0	0
79	cvt.f64.f32	2	8				4			0	144	0	0
80	add.s32	2	2				1			2	81	1	0
81	setp.gt.s32	2	16+1				1			0	85	0	0
82	sub.s32	2	2				1			2	83	1	0
83	add.s32	2	16+1				1			17	84	1	0
84	add.s32	2	16+1				1			17	85	1	0
85	selp.b32	2	16+1				1			17	86	1	0
86	setp.lt.s32	2	16+1				1			0	88	0	0
87	add.s32	2	2				1			2	88	1	0
88	selp.b32	2	16+1				1			17	89	1	0
89	mul.wide.s32	2	16+1				1			0	92	0	0
90	mul.wide.s32	2	2				1			2	91	1	0
91	add.s64	2	16+1				1			17	92	1	0
92	add.s64	2	16+1				1			0	151	0	0
93	cvt.f64.f32	2	8				4			0	155	0	0
94	mov.u32	2	2				1			2	95	1	0
95	sub.s32	2	16+1				1			0	169	0	0
96	mov.u32	2	2				1			1	98	0	0
--	BB0_4:									--	--	--	--
97	mov.u32	2	16+1				1			17	98	1	0
98	sub.s32	2	16+1				1			17	99	1	0
99	setp.le.s32	2	16+1				1			0	102	0	0
100	add.s32	2	2				1			2	101	1	0
101	setp.ge.s32	2	16+1				1			17	102	1	0
102	and.pred	2	16+1				1			0	104	0	0
103	mov.u16	2	2				1			0	171	0	0
104	@!%p15 bra BB0_7;	2	2				1			1	0	0	0
105	bra.uni BB0_5;	2	2				1			0	0	0	0
--	BB0_5:									--	--	--	--
106	setp.gt.s32	2	2				1			0	108	0	0
107	setp.lt.s32	2	2				1			2	108	1	0
108	or.pred	2	16+1				1			0	115	0	0
109	shr.s32	2	2				1			0	112	0	0
110	setp.gt.s32	2	2				1			0	114	0	0
111	neg.s32	2	2				1			2	112	1	0
112	and.b32	2	16+1				1			17	113	1	0
113	setp.lt.s32	2	16+1				1			17	114	1	0
114	or.pred	2	16+1				1			17	115	1	0
115	or.pred	2	16+1				1			0	122	0	0
116	shr.s32	2	2				1			0	119	0	0
117	setp.gt.s32	2	2				1			0	121	0	0
118	neg.s32	2	2				1			2	119	1	0
119	and.b32	2	16+1				1			17	120	1	0
120	setp.lt.s32	2	16+1				1			17	121	1	0
121	or.pred	2	16+1				1			17	122	1	0
122	or.pred	2	16+1				1			17	123	1	0
123	@%p28 bra BB0_7;	2	16+1				1			1	0	0	0
124	add.s64	2	2				1			2	125	1	1
125	ld.shared.f32	4				16+2	2			18	126	1	1
126	cvt.f64.f32	2	16+4				4			0	142	0	0
127	mov.u64	2	2				1			2	128	1	0
128	add.s64	2	16+1				1			17	129	1	0
129	add.s64	2	16+1				1			17	130	1	1
130	ld.shared.f32	4				16+2	2			18	131	1	1
131	cvt.f64.f32	2	16+4				4			0	144	0	0
132	setp.gt.s32	2	2				1			0	134	0	0
133	add.s32	2	2				1			2	134	1	0

134	selp.b32	2	16+1				1			17	135	1	0
135	mul.wide.s32	2	16+1				1			17	136	1	0
136	add.s64	2	16+1				1			17	137	1	0
137	add.s64	2	16+1				1			17	138	1	1
138	ld.shared.f32	4				16+2	2			0	140	0	0
139	ld.shared.f32	4				4	2			4	140	1	1
140	add.f32	2	16+1				1			17	141	1	0
141	cvt.f64.f32		16+4				4			0	143	0	1
142	add.f64	8		16+4			4			20	143	1	0
143	sub.f64	8		8			4			8	144	1	0
144	fma.rn.f64	8		46+4			4			0	155	0	1
145	setp.gt.s32		2				1			0	147	0	0
146	add.s32	2	2				1			2	147	1	0
147	selp.b32	2	16+1				1			17	148	1	0
148	mul.wide.s32	2	16+1				1			17	149	1	0
149	add.s64	2	16+1				1			17	150	1	1
150	ld.shared.f32	4				16+2	2			0	152	0	0
151	ld.shared.f32	4				4	2			4	152	1	1
152	add.f32	2	16+1				1			17	153	1	0
153	cvt.f64.f32	2	16+4				4			20	154	1	0
154	sub.f64	8		16+4			4			20	155	1	0
155	fma.rn.f64	8		46+4			4			0	160	0	1
156	mov.f32		2				1			2	157	1	0
157	sub.f32	2	16+1				1			17	158	1	0
158	mul.f32	2	16+1				1			17	159	1	0
159	cvt.f64.f32	2	16+4				4			20	160	1	0
160	add.f64	8		8			4			8	161	1	0
161	fma.rn.f64	8		46+4			4			50	162	1	1
162	cvt.rn.f32.f64	2	16+4				4			0	166	0	0
163	mov.u64	2	16+1				1			17	164	1	0
164	add.s64	2	16+1				1			17	165	1	0
165	add.s64	2	16+1				1			17	166	1	1
166	st.shared.f32	4				41+2	2			0	0	0	1
167	mov.u16		2				1			0	171	0	0
--	BB0_7:									--	--	--	--
168	bar.sync	2							173	1	0	0	0
169	setp.eq.s32	2	16+1				1			17	170	1	0
170	@%p31 bra BB0_11;	2	16+1				1			1	0	0	0
171	setp.eq.s16	2	2				1			2	172	1	0
172	@%p32 bra BB0_10;	2	16+1				1			1	0	0	0
173	mov.u64	2	2				1			2	174	1	0
174	add.s64	2	16+1				1			17	175	1	0
175	add.s64	2	16+1				1			17	176	1	1
176	ld.shared.f32	4				16+2	2			0	178	0	1
177	add.s64		2				1			2	178	1	1
178	st.shared.f32	4				41+2	2			0	0	0	0
--	BB0_10:									--	--	--	--
179	bar.sync	2							173	1	0	0	0
180	add.s32	2	16+1				1			0	0	0	0
181	setp.lt.s32	2	2				1			2	182	1	0
182	@%p33 bra BB0_4;	2	16+1				1			1	0	0	0
--	BB0_11:									--	--	--	--
183	and.b16	2	2				1			2	184	1	0
184	setp.eq.s16	2	16+1				1			17	185	1	0
185	@%p34 bra BB0_13;	2	16+1				1			1	0	0	0
186	mul.wide.s32	2	2				1			0	188	0	0
187	mov.u64	2	2				1			2	188	1	0
188	add.s64	2	16+1				1			0	190	0	0
189	mul.wide.s32	2	2				1			2	190	1	0
190	add.s64	2	16+1				1			17	191	1	1
191	ld.shared.f32	4				16+2	2			0	196	0	1
192	mad.lo.s32		2				1			0	194	0	0
193	cvta.to.global.u64	2	2				1			0	195	0	0
194	mul.wide.s32	2	2				1			2	195	1	0
195	add.s64	2	16+1				1			17	196	1	1
196	st.global.f32	4					--	764		1	0	0	0
--	BB0_13:												
--	ret;												

Table 2-3, Level 1 and level 2 supersteps for Hotspot on GTX 760 (s_1=4, s_2=119)

Level1: Superstep#: <start ~ end, comp, comm, ovh>, Itteration

1: <1~96,1578,1528,173>, 1 2: <97~170,1168,0,173>, 2

3: <171~182,174,0,173>, 1 4: <183~196,133,764,0>, 1

Level2: Superstep#: <start ~ end, comp, comm, ovh>

```
1: <1~16,49,0,0>       31: <65~66,21,0,0>     61: <115~118,25,0,0>   91: <157~157,19,0,0>
2: <17~18,21,0,0>      32: <67~67,19,0,0>     62: <119~119,19,0,0>   92: <158~158,19,0,0>
3: <19~19,19,0,0>      33: <68~68,19,0,0>     63: <120~120,19,0,0>   93: <159~159,22,0,0>
4: <20~21,21,0,0>      34: <69~69,19,0,0>     64: <121~121,19,0,0>   94: <160~160,16,0,0>
5: <22~23,21,0,0>      35: <70~70,19,0,0>     65: <122~122,19,0,0>   95: <161~161,58,0,0>
6: <24~25,21,0,0>      36: <71~72,21,0,0>     66: <123~123,19,0,0>   96: <162~163,39,0,0>
7: <26~28,23,0,0>      37: <73~73,19,0,0>     67: <124~124,4,0,0>    97: <164~164,19,0,0>
8: <29~29,19,0,0>      38: <74~75,21,0,0>     68: <125~125,22,0,0>   98: <165~165,19,0,0>
9: <30~32,23,0,0>      39: <76~77,21,0,0>     69: <126~127,24,0,0>   99: <166~168,47,0,173>
10: <33~33,19,0,0>     40: <78~80,29,0,0>     70: <128~128,19,0,0>   100: <169~169,19,0,0>
11: <34~34,19,0,0>     41: <81~82,21,0,0>     71: <129~129,19,0,0>   101: <170~170,19,0,0>
12: <35~35,19,0,0>     42: <83~83,19,0,0>     72: <130~130,22,0,0>   102: <171~171,4,0,0>
13: <36~36,19,0,0>     43: <84~84,19,0,0>     73: <131~133,26,0,0>   103: <172~172,19,0,0>
14: <37~38,21,0,0>     44: <85~85,19,0,0>     74: <134~134,19,0,0>   104: <173~173,4,0,0>
15: <39~40,6,0,0>      45: <86~87,21,0,0>     75: <135~135,19,0,0>   105: <174~174,19,0,0>
16: <41~41,19,0,0>     46: <88~88,19,0,0>     76: <136~136,19,0,0>   106: <175~175,19,0,0>
17: <42~43,6,764,0>    47: <89~90,21,0,0>     77: <137~137,19,0,0>   107: <176~177,22,0,0>
18: <44~44,4,0,0>      48: <91~91,19,0,0>     78: <138~139,26,0,0>   108: <178~179,47,0,173>
19: <45~46,21,0,0>     49: <92~94,29,0,0>     79: <140~140,19,0,0>   109: <180~181,21,0,0>
20: <47~47,19,0,0>     50: <95~96,21,0,0>     80: <141~142,20,0,0>   110: <182~182,19,0,0>
21: <48~49,47,0,0>     51: <97~97,19,0,0>     81: <143~143,16,0,0>   111: <183~183,4,0,0>
22: <50~50,19,0,0>     52: <98~98,19,0,0>     82: <144~146,58,0,0>   112: <184~184,19,0,0>
23: <51~52,6,764,0>    53: <99~100,21,0,0>    83: <147~147,19,0,0>   113: <185~185,19,0,0>
24: <53~53,19,0,0>     54: <101~101,19,0,0>   84: <148~148,19,0,0>   114: <186~187,6,0,0>
25: <54~54,19,0,0>     55: <102~104,23,0,0>   85: <149~149,19,0,0>   115: <188~189,21,0,0>
26: <55~56,47,0,173>   56: <105~107,8,0,0>    86: <150~151,26,0,0>   116: <190~190,19,0,0>
27: <57~57,19,0,0>     57: <108~111,25,0,0>   87: <152~152,19,0,0>   117: <191~194,22,0,0>
28: <58~58,19,0,0>     58: <112~112,19,0,0>   88: <153~153,22,0,0>   118: <195~195,19,0,0>
29: <59~59,145,0,0>    59: <113~113,19,0,0>   89: <154~154,28,0,0>   119: <196~196,4,764,0>
30: <60~64,433,0,0>    60: <114~114,19,0,0>   90: <155~156,58,0,0>
```

$parallel\_comp\ (eq.10) = 1 \times 1578 + 2 \times 1168 + 1 \times 174 + 1 \times 133 = 4221$

$block\_bar\_ovh\ (eq.11) = 1 \times 173 + 2 \times 173 + 1 \times 173 + 1 \times 0 = 692$

$block\_comm\ (eq.12) = 1 \times 1528 + 2 \times 0 + 1 \times 0 + 1 \times 764 = 2292$

$block\_comm_{\Delta} = 1528$

$COMP\ (eq.5) = 2 \times 10 + 4221 = 4241$

$warp\_comp = COMP \div w = 4241 \div 2 = 2121$

$warp\_comm_{\Delta} = 764$

$$warps\_need\ (eq.15) = 4 \times \left(\left\lceil \frac{764 \times 265}{2121 \times (3-1)} \right\rceil + 1\right) = 196$$

$$nonoverlapped\ (eq.17) = \min\left\{\frac{2292}{2}, 191 + \frac{1528}{2} \times \max\left(0{,}1 - \frac{2 \times 4}{196}\right)\right\} = 924$$

$block\_exec\_cycle\ (eq.4) = 553 + 692 + 924 + 4241 = 6410$

$comp\ (eq.19) = 4933$

$novlp\ (eq.20) = 924$

$$\rho\ (eq.21) = max\left\{1, min\left\{\left\lfloor \frac{2048}{256} \right\rfloor, \left\lfloor \frac{65536}{\lfloor 256 \times 34 \rfloor} \right\rfloor, \left\lfloor \frac{48\ KB}{3072\ B} \right\rfloor\right\}\right\} = 7$$

$$\mathrm{K}\ (eq.22) = \frac{1849}{6 \times 7} = 44.023$$

$$\tau\ (eq.23) = \left\lceil \frac{924}{4933} \right\rceil + 1 = 2$$

$\mu = 3.36$

$$\rho \geq \tau : kernel\_exec\_cycle\ (eq.24) = 553 + \frac{1849}{6} \times \frac{4933}{3.36} + \frac{924}{2} = 453452$$

$mesured\_kernel\_exec\_cycle = 475105$

$$error\ \% = \frac{|475105 - 453452|}{475105} \times 100 = 4.55\ \%$$

3. Details of results for KNN on GTX 760

$d = 3$	$n_b = 168$	$g_0 = 32$	$\max\_thread\_per\_sm = 2048$
$n_c = 192$	$n_t = 256$	$g_1 = 98$	$reg\_per\_thread = 9$
$n_{SM} = 6$	$h = 28$	$g_2 = 61$	$shmem\_size = 48\ KB$
$n_{ws} = 4$	$l = 28$	$mem\_lat = G_2 = 191$	$shmem\_per\_block = 0\ B$
$n_{du} = 8$	$l_c = 26$	$warp\_lnch\_ovh = 10$	$n_{reg} = 65536$
$n_{fu} = 4$	$l_m = 2$	$block\_lnch\_ovh = 553$	
$warp\_size = 32$	$l_b = 0$	$issue\_cycle\ = 1$	

$$w\ (eq.\ 6) = \left\lceil \frac{256}{32 \times 4} \right\rceil = 2$$

Table 3-1, Latency, FUs, Throughput, u, d and p for KNN on GTX 760

Inst#	Instructions	FU	n_{fu}^i	T_i	n_{ma}^i	Latency		u	d	p
						comp	comm			
1	ld.param.u64 %rd1, [Kernel_param_0];	SPs	32	32		16		15	0	0
2	ld.param.u64 %rd2, [Kernel_param_1];	SPs	32	32		16		16	0	0
3	ld.param.u32 %r2, [Kernel_param_2];	SPs	32	32		16		13	0	0
4	ld.param.f32 %f1, [Kernel_param_3];	SPs	32	32		16		22	0	0
5	ld.param.f32 %f2, [Kernel_param_4];	SPs	32	32		16		24	0	0
6	mov.u32 %r3, %ctaid.y;	SPs	32	32		32		9	0	0
7	mov.u32 %r4, %nctaid.x;	SPs	32	32		16		9	0	0
8	mov.u32 %r5, %ctaid.x;	SPs	32	32		32		9	1	0
9	mad.lo.s32 %r6, %r3, %r4, %r5;	SPs	32	32		16		12	0	0
10	mov.u32 %r7, %ntid.x;	SPs	32	32		16		12	0	0
11	mov.u32 %r8, %tid.x;	SPs	32	32		32		12	0	0
12	mad.lo.s32 %r1, %r6, %r7, %r8;	SPs	32	32		16		13	1	0
13	setp.ge.s32 %p1, %r1, %r2;	SPs	32	32		16		14	1	0
14	@%p1 bra BB0_2;	SPs	32	32		16		0	0	0
15	cvta.to.global.u64 %rd3, %rd1;	SPs	32	32		16		20	0	0
16	cvta.to.global.u64 %rd4, %rd2;	SPs	32	32		16		18	0	0
17	mul.wide.s32 %rd5, %r1, 4;	SPs	32	32		16		18	1	0
18	add.s64 %rd6, %rd4, %rd5;	SPs	32	32		16		28	0	0
19	mul.wide.s32 %rd7, %r1, 8;	SPs	32	32		16		20	1	0
20	add.s64 %rd8, %rd3, %rd7;	SPs	32	32		16		21	1	1
21	ld.global.f32 %f3, [%rd8];	LDST	16	16	2		191	22	1	1
22	sub.f32 %f4, %f1, %f3;	SPs	32	32		16		26	0	1
23	ld.global.f32 %f5, [%rd8+4];	LDST	16	16	2	32[1]		24	1	1
24	sub.f32 %f6, %f2, %f5;	SPs	32	32		16		25	1	0
25	mul.f32 %f7, %f6, %f6;	SPs	32	32		16		26	1	0
26	fma.rn.f32 %f8, %f4, %f4, %f7;	SPs	32	32		41		27	1	1
27	sqrt.rn.f32 %f9, %f8;	SFU	8	8		411		28	1	1
28	st.global.f32 [%rd6], %f9;	LDST	16	16	2		191	0	--	--
--	BB0_2:									
--	ret;									

Table 3-2, BLOCK data structure for KNN on GTX 760

Inst#	Instructions	$comp_i$						$comm_i$	ovh_i	$sync_i$	u	d	p
		Issue	FU_1	FU_2	FU_3	FU_4	$32/T_i$						
1	ld.param.u64	2	16+1				1			0	15	0	0
2	ld.param.u64	2	2				1			0	16	0	0
3	ld.param.u32	2	2				1			0	13	0	0
4	ld.param.f32	2	2				1			0	22	0	0
5	ld.param.f32	2	2				1			0	24	0	0
6	mov.u32	2	2				1			0	9	0	0
7	mov.u32	2	2				1			0	9	0	0
8	mov.u32	2	2				1			2	9	1	0
9	mad.lo.s32	2	16+1				1			0	12	0	0
10	mov.u32	2	2				1			0	12	0	0
11	mov.u32	2	2				1			2	12	0	0
12	mad.lo.s32	2	2				1			2	13	1	0
13	setp.ge.s32	2	16+1				1			17	14	1	0
14	@%p1 bra BB0_2;	2	16+1				1			1	0	0	0
15	cvta.to.global.u64	2	2				1			0	20	0	0
16	cvta.to.global.u64	2	2				1			0	18	0	0

[1] L1 cache hit

17	mul.wide.s32	2	2				1			2	18	1	0
18	add.s64	2	16+1				1			0	28	0	0
19	mul.wide.s32	2	2				1			2	20	1	0
20	add.s64	2	16+1				1			17	21	1	1
21	ld.global.f32	4					2	764		1	22	1	1
22	sub.f32		16+1				1			0	26	0	1
23	ld.global.f32	4				32+6	2			38	24	1	1
24	sub.f32	2	16+1				1			17	25	1	0
25	mul.f32	2	16+1				1			17	26	1	0
26	fma.rn.f32	2	41+1				1			42	27	1	1
27	sqrt.rn.f32	8			411+4		4			415	28	1	1
28	st.global.f32	4					2	764		1	0	--	--
--	BB0_2:												
--	ret;												

Table 3-3, Level 1 and Level 2 supersteps for KNN on GTX 760 (s_1=2, s_2=15)

Level1:	Superstep#: <start ~ end, comp, comm, ovh>, Itteration		
	1: <1~14,98,0,0>, 1	2: <15~28,599,1528,0>, 1	
Level2:	Superstep#: <start ~ end, comp, comm, ovh>		
1: <1~8,33,0,0>	5: <14~14,19,0,0>	9: <21~21,4,764,0>	13: <26~26,44,0,0>
2: <9~11,23,0,0>	6: <15~17,8,0,0>	10: <22~23,38,0,0>	14: <27~27,423,0,0>
3: <12~12,4,0,0>	7: <18~19,21,0,0>	11: <24~24,19,0,0>	15: <28~28,4,764,0>
4: <13~13,19,0,0>	8: <20~20,19,0,0>	12: <25~25,19,0,0>	

$parallel\_comp\ (eq.10) = 1 \times 98 + 1 \times 599 = 697$

$block\_bar\_ovh\ (eq.11) = 0$

$block\_comm\ (eq.12) = 1528$

$block\_comm_{\Delta} = 764$

$COMP\ (eq.5) = 2 \times 10 + 697 = 717$

$warp\_comp = COMP \div w = 717 \div 2 = 359$

$warp\_comm_{\Delta} = 382$

$$warps\_need\ (eq.15) = 4 \times \left(\left\lceil \frac{382 \times 26}{359 \times (2-1)} \right\rceil + 1\right) = 116$$

$$nonoverlapped\ (eq.17) = \min\left\{\frac{1528}{2}, 191 + \frac{764}{2} \times \max\left(0{,}1 - \frac{2 \times 4}{116}\right)\right\} = 547$$

$block\_exec\_cycle\ (eq.4) = 553 + 0 + 547 + 717 = 1817$

$comp\ (eq.19) = 717$

$novlp\ (eq.20) = 547$

$$\rho\ (eq.21) = max\left\{1, min\left\{\left\lfloor \frac{2048}{256} \right\rfloor, \left\lfloor \frac{65536}{256 \times 9} \right\rfloor, \left\lfloor \frac{48\ KB}{0\ B} \right\rfloor\right\}\right\} = 8$$

$$\mathrm{K}\ (eq.22) = \frac{168}{6 \times 8} = 3.5$$

$$\tau\ (eq.23) = \left\lceil \frac{547}{717} \right\rceil + 1 = 2$$

$\mu = 3.36$

$$\rho \geq \tau : kernel\_exec\_cycle\ (eq.24) = 553 + \frac{168}{6} \times \frac{717}{3.36} + \frac{547}{2} = 6802$$

$mesured\_kernel\_exec\_cycle = 7458$

$$error\ \% = \frac{|7458 - 6802|}{7458} \times 100 = 8.79\ \%$$

4. Details of results for MM on GTX 760

$d = 3$	$n_b = 200$	$g_0 = 32$	$\max\_thread\_per\_sm = 2048$
$n_c = 192$	$n_t = 1024$	$g_1 = 98$	$reg\_per\_thread = 22$
$n_{SM} = 6$	$h = 155$	$g_2 = 61$	$shmem\_size = 48\ KB$
$n_{ws} = 4$	$l = 1163$	$mem\_lat = G_2 = 191$	$shmem\_per\_block = 2048\ B$
$n_{du} = 8$	$l_c = 1122$	$warp\_lnch\_ovh = 10$	$n_{reg} = 65536$
$n_{fu} = 4$	$l_m = 21$	$block\_lnch\_ovh = 553$	
$warp\_size = 32$	$l_b = 20$	$issue\_cycle\ = 1$	

$$w\ (eq.\ 6) = \left\lceil \frac{1024}{32 \times 4} \right\rceil = 8$$

Table 4-1, Latency, FUs, Throughput, u, d and p for MM on GTX 760

Inst#	Instructions	FU	n_{fu}^i	T_i	n_{ma}^i	Latency (Cycle)		u	d	p
						محاسبات	ارتباطات			
1	ld.param.u64 %rd7, [Kernel_param_0];	SPs	32	32		16		152	0	0
2	ld.param.u64 %rd8, [Kernel_param_1];	SPs	32	32		16		15	0	0
3	ld.param.u64 %rd9, [Kernel_param_2];	SPs	32	32		16		14	0	0
4	ld.param.u32 %r12, [Kernel_param_3];	SPs	32	32		16		7	0	0
5	ld.param.u32 %r13, [Kernel_param_4];	SPs	32	32		16		10	0	0
6	mov.u32 %r14, %ctaid.y;	SPs	32	32		32		7	1	0
7	mul.lo.s32 %r15, %r12, %r14;	SPs	32	32		16		8	1	0
8	shl.b32 %r30, %r15, 5;	SPs	32	32		16		17	0	0
9	add.s32 %r2, %r12, -1;	SPs	32	32		16		11	0	0
10	shl.b32 %r3, %r13, 5;	SPs	32	32		16		141	0	0
11	setp.lt.s32 %p1, %r2, 0;	SPs	32	32		16		13	0	0
12	mov.f32 %f103, 0f00000000;	SPs	32	32		16		32	0	0
13	@%p1 bra BB0_3;	SPs	32	32		16		0	0	0
14	cvta.to.global.u64 %rd1, %rd9;	SPs	32	32		16		40	0	0
15	cvta.to.global.u64 %rd2, %rd8;	SPs	32	32		16		35	0	0
16	mov.u32 %r16, %tid.x;	SPs	32	32		32		21	0	0
17	add.s32 %r4, %r2, %r30;	SPs	32	32		16		143	0	0
18	mov.u32 %r17, %ctaid.x;	SPs	32	32		32		19	1	0
19	shl.b32 %r29, %r17, 5;	SPs	32	32		16		38	0	0
20	mov.u32 %r18, %tid.y;	SPs	32	32		32		21	1	0
21	mad.lo.s32 %r6, %r18, %r12, %r16;	SPs	32	32		16		33	0	0
22	mul.wide.s32 %rd10, %r18, 128;	SPs	32	32		16		24	0	0
23	mov.u64 %rd11, Kernel $__cuda_...;	SPs	32	32		16		24	1	0
24	add.s64 %rd5, %rd11, %rd10;	SPs	32	32		16		26	0	0
25	mul.wide.s32 %rd12, %r16, 4;	SPs	32	32		16		26	1	0
26	add.s64 %rd3, %rd5, %rd12;	SPs	32	32		16		37	0	0
27	mad.lo.s32 %r7, %r18, %r13, %r16;	SPs	32	32		16		38	0	0
28	mov.u64 %rd13, Kernel $__cuda_...;	SPs	32	32		16		29	1	0
29	add.s64 %rd14, %rd13, %rd10;	SPs	32	32		16		30	1	0
30	add.s64 %rd4, %rd14, %rd12;	SPs	32	32		16		42	0	0
31	add.s64 %rd6, %rd13, %rd12;	SPs	32	32		16		44	0	0
32	mov.f32 %f103, 0f00000000;	SPs	32	32		16		46	0	0
--	BB1_2:							--	--	--
33	add.s32 %r19, %r6, %r30;	SPs	32	32		16		34	1	0
34	mul.wide.s32 %rd15, %r19, 4;	SPs	32	32		16		35	1	0
35	add.s64 %rd16, %rd2, %rd15;	SPs	32	32		16		36	1	1
36	ld.global.f32 %f6, [%rd16];	LDST	16	16	1		191	37	1	0
37	st.shared.f32 [%rd3], %f6;	LDST	16	16	1	41		0	0	1
38	add.s32 %r20, %r7, %r29;	SPs	32	32		16		39	1	0
39	mul.wide.s32 %rd17, %r20, 4;	SPs	32	32		16		40	1	0
40	add.s64 %rd18, %rd1, %rd17;	SPs	32	32		16		41	1	1
41	ld.global.f32 %f7, [%rd18];	LDST	16	16	1		191	42	1	0
42	st.shared.f32 [%rd4], %f7;	LDST	16	16	1	41		0	0	0
43	bar.sync 0;		MI			297		0	0	0
44	ld.shared.f32 %f8, [%rd6];	LDST	16	16	1	16		46	0	0
45	ld.shared.f32 %f9, [%rd5];	LDST	16	16	1	16		46	1	1
46	fma.rn.f32 %f10, %f9, %f8, %f103;	SPs	32	32		41		49	0	1
47	ld.shared.f32 %f11, [%rd6+128];	LDST	16	16	1	16		49	0	0
48	ld.shared.f32 %f12, [%rd5+4];	LDST	16	16	1	16		49	1	1
49	fma.rn.f32 %f13, %f12, %f11, %f10;	SPs	32	32		41		52	0	1
50	ld.shared.f32 %f14, [%rd6+256];	LDST	16	16	1	16		52	0	0
51	ld.shared.f32 %f15, [%rd5+8];	LDST	16	16	1	16		52	1	1
52	fma.rn.f32 %f16, %f15, %f14, %f13;	SPs	32	32		41		55	0	1
53	ld.shared.f32 %f17, [%rd6+384];	LDST	16	16	1	16		55	0	0

54	ld.shared.f32	%f18, [%rd5+12];	LDST	16	16	1	16		55	1	1
55	fma.rn.f32	%f19, %f18, %f17, %f16;	SPs	32	32		41		58	0	1
56	ld.shared.f32	%f20, [%rd6+512];	LDST	16	16	1	16		58	0	0
57	ld.shared.f32	%f21, [%rd5+16];	LDST	16	16	1	16		58	1	1
58	fma.rn.f32	%f22, %f21, %f20, %f19;	SPs	32	32		41		61	0	1
59	ld.shared.f32	%f23, [%rd6+640];	LDST	16	16	1	16		61	0	0
60	ld.shared.f32	%f24, [%rd5+20];	LDST	16	16	1	16		61	1	1
61	fma.rn.f32	%f25, %f24, %f23, %f22;	SPs	32	32		41		64	0	1
62	ld.shared.f32	%f26, [%rd6+768];	LDST	16	16	1	16		64	0	0
63	ld.shared.f32	%f27, [%rd5+24];	LDST	16	16	1	16		64	1	1
64	fma.rn.f32	%f28, %f27, %f26, %f25;	SPs	32	32		41		67	0	1
65	ld.shared.f32	%f29, [%rd6+896];	LDST	16	16	1	16		67	0	0
66	ld.shared.f32	%f30, [%rd5+28];	LDST	16	16	1	16		67	1	1
67	fma.rn.f32	%f31, %f30, %f29, %f28;	SPs	32	32		41		70	0	1
68	ld.shared.f32	%f32, [%rd6+1024];	LDST	16	16	1	16		70	0	0
69	ld.shared.f32	%f33, [%rd5+32];	LDST	16	16	1	16		70	1	1
70	fma.rn.f32	%f34, %f33, %f32, %f31;	SPs	32	32		41		73	0	1
71	ld.shared.f32	%f35, [%rd6+1152];	LDST	16	16	1	16		73	0	0
72	ld.shared.f32	%f36, [%rd5+36];	LDST	16	16	1	16		73	1	1
73	fma.rn.f32	%f37, %f36, %f35, %f34;	SPs	32	32		41		76	0	1
74	ld.shared.f32	%f38, [%rd6+1280];	LDST	16	16	1	16		76	0	0
75	ld.shared.f32	%f39, [%rd5+40];	LDST	16	16	1	16		76	1	1
76	fma.rn.f32	%f40, %f39, %f38, %f37;	SPs	32	32		41		79	0	1
77	ld.shared.f32	%f41, [%rd6+1408];	LDST	16	16	1	16		79	0	0
78	ld.shared.f32	%f42, [%rd5+44];	LDST	16	16	1	16		79	1	1
79	fma.rn.f32	%f43, %f42, %f41, %f40;	SPs	32	32		41		82	0	1
80	ld.shared.f32	%f44, [%rd6+1536];	LDST	16	16	1	16		82	0	0
81	ld.shared.f32	%f45, [%rd5+48];	LDST	16	16	1	16		82	1	1
82	fma.rn.f32	%f46, %f45, %f44, %f43;	SPs	32	32		41		85	0	1
83	ld.shared.f32	%f47, [%rd6+1664];	LDST	16	16	1	16		85	0	0
84	ld.shared.f32	%f48, [%rd5+52];	LDST	16	16	1	16		85	1	1
85	fma.rn.f32	%f49, %f48, %f47, %f46;	SPs	32	32		41		88	0	1
86	ld.shared.f32	%f50, [%rd6+1792];	LDST	16	16	1	16		88	0	0
87	ld.shared.f32	%f51, [%rd5+56];	LDST	16	16	1	16		88	1	1
88	fma.rn.f32	%f52, %f51, %f50, %f49;	SPs	32	32		41		91	0	1
89	ld.shared.f32	%f53, [%rd6+1920];	LDST	16	16	1	16		91	0	0
90	ld.shared.f32	%f54, [%rd5+60];	LDST	16	16	1	16		91	1	1
91	fma.rn.f32	%f55, %f54, %f53, %f52;	SPs	32	32		41		94	0	1
92	ld.shared.f32	%f56, [%rd6+2048];	LDST	16	16	1	16		94	0	0
93	ld.shared.f32	%f57, [%rd5+64];	LDST	16	16	1	16		94	1	1
94	fma.rn.f32	%f58, %f57, %f56, %f55;	SPs	32	32		41		97	0	1
95	ld.shared.f32	%f59, [%rd6+2176];	LDST	16	16	1	16		97	0	0
96	ld.shared.f32	%f60, [%rd5+68];	LDST	16	16	1	16		97	1	1
97	fma.rn.f32	%f61, %f60, %f59, %f58;	SPs	32	32		41		100	0	1
98	ld.shared.f32	%f62, [%rd6+2304];	LDST	16	16	1	16		100	0	0
99	ld.shared.f32	%f63, [%rd5+72];	LDST	16	16	1	16		100	1	1
100	fma.rn.f32	%f64, %f63, %f62, %f61;	SPs	32	32		41		103	0	1
101	ld.shared.f32	%f65, [%rd6+2432];	LDST	16	16	1	16		103	0	0
102	ld.shared.f32	%f66, [%rd5+76];	LDST	16	16	1	16		103	1	1
103	fma.rn.f32	%f67, %f66, %f65, %f64;	SPs	32	32		41		106	0	1
104	ld.shared.f32	%f68, [%rd6+2560];	LDST	16	16	1	16		106	0	0
105	ld.shared.f32	%f69, [%rd5+80];	LDST	16	16	1	16		106	1	1
106	fma.rn.f32	%f70, %f69, %f68, %f67;	SPs	32	32		41		109	0	1
107	ld.shared.f32	%f71, [%rd6+2688];	LDST	16	16	1	16		109	0	0
108	ld.shared.f32	%f72, [%rd5+84];	LDST	16	16	1	16		109	1	1
109	fma.rn.f32	%f73, %f72, %f71, %f70;	SPs	32	32		41		112	0	1
110	ld.shared.f32	%f74, [%rd6+2816];	LDST	16	16	1	16		112	0	0
111	ld.shared.f32	%f75, [%rd5+88];	LDST	16	16	1	16		112	1	1
112	fma.rn.f32	%f76, %f75, %f74, %f73;	SPs	32	32		41		115	0	1
113	ld.shared.f32	%f77, [%rd6+2944];	LDST	16	16	1	16		115	0	0
114	ld.shared.f32	%f78, [%rd5+92];	LDST	16	16	1	16		115	1	1
115	fma.rn.f32	%f79, %f78, %f77, %f76;	SPs	32	32		41		118	0	1
116	ld.shared.f32	%f80, [%rd6+3072];	LDST	16	16	1	16		118	0	0
117	ld.shared.f32	%f81, [%rd5+96];	LDST	16	16	1	16		118	1	1
118	fma.rn.f32	%f82, %f81, %f80, %f79;	SPs	32	32		41		121	0	1
119	ld.shared.f32	%f83, [%rd6+3200];	LDST	16	16	1	16		121	0	0
120	ld.shared.f32	%f84, [%rd5+100];	LDST	16	16	1	16		121	1	1
121	fma.rn.f32	%f85, %f84, %f83, %f82;	SPs	32	32		41		124	0	1
122	ld.shared.f32	%f86, [%rd6+3328];	LDST	16	16	1	16		124	0	0
123	ld.shared.f32	%f87, [%rd5+104];	LDST	16	16	1	16		124	1	1
124	fma.rn.f32	%f88, %f87, %f86, %f85;	SPs	32	32		41		127	0	1
125	ld.shared.f32	%f89, [%rd6+3456];	LDST	16	16	1	16		127	0	0
126	ld.shared.f32	%f90, [%rd5+108];	LDST	16	16	1	16		127	1	1
127	fma.rn.f32	%f91, %f90, %f89, %f88;	SPs	32	32		41		130	0	1

128	ld.shared.f32 %f92, [%rd6+3584];	LDST	16	16	1	16		130	0	0
129	ld.shared.f32 %f93, [%rd5+112];	LDST	16	16	1	16		130	1	1
130	fma.rn.f32 %f94, %f93, %f92, %f91;	SPs	32	32		41		133	0	1
131	ld.shared.f32 %f95, [%rd6+3712];	LDST	16	16	1	16		133	0	0
132	ld.shared.f32 %f96, [%rd5+116];	LDST	16	16	1	16		133	1	1
133	fma.rn.f32 %f97, %f96, %f95, %f94;	SPs	32	32		41		136	0	1
134	ld.shared.f32 %f98, [%rd6+3840];	LDST	16	16	1	16		136	0	0
135	ld.shared.f32 %f99, [%rd5+120];	LDST	16	16	1	16		136	1	1
136	fma.rn.f32 %f100, %f99, %f98, %f97;	SPs	32	32		41		139	0	1
137	ld.shared.f32 %f101, [%rd6+3968];	LDST	16	16	1	16		139	0	0
138	ld.shared.f32 %f102, [%rd5+124];	LDST	16	16	1	16		139	1	1
139	fma.rn.f32 %f103, %f102, %f101, %f100;	SPs	32	32		41		155	0	0
140	bar.sync 0;		MI			297		0	0	0
141	add.s32 %r29, %r29, %r3;	SPs	32	32		16		0	0	0
142	add.s32 %r30, %r30, 32;	SPs	32	32		16		143	1	0
143	setp.le.s32 %p2, %r30, %r4;	SPs	32	32		16		144	1	0
144	@%p2 bra BB1_2;	SPs	32	32		16		0	0	0
--	BB1_3:									
145	mov.u32 %r22, %ctaid.x;	SPs	32	32		32		146	1	0
146	shl.b32 %r23, %r22, 5;	SPs	32	32		16		147	1	0
147	mad.lo.s32 %r24, %r14, %r3, %r23;	SPs	32	32		16		149	0	0
148	mov.u32 %r25, %tid.x;	SPs	32	32		32		149	1	0
149	add.s32 %r26, %r24, %r25;	SPs	32	32		16		151	0	0
150	mov.u32 %r27, %tid.y;	SPs	32	32		32		151	1	0
151	mad.lo.s32 %r28, %r27, %r13, %r26;	SPs	32	32		16		153	0	1
152	cvta.to.global.u64 %rd19, %rd7;	SPs	32	32		16		154	0	1
153	mul.wide.s32 %rd20, %r28, 4;	SPs	32	32		16		154	1	0
154	add.s64 %rd21, %rd19, %rd20;	SPs	32	32		16		155	1	1
155	st.global.f32 [%rd21], %f103;	LDST	16	16	1		191	0	--	--
--	ret;									

Table 4-2, BLOCK data structure for MM on GTX 760

Inst#	Instructions	$comp_i$						$comm_i$	ovh_i	$sync_i$	u	d	p
		Issue	FU_1	FU_2	FU_3	FU_4	$32/T_i$						
1	ld.param.u64	8	16+7				1			0	152	0	0
2	ld.param.u64	8	8				1			0	15	0	0
3	ld.param.u64	8	8				1			0	14	0	0
4	ld.param.u32	8	8				1			0	7	0	0
5	ld.param.u32	8	8				1			0	10	0	0
6	mov.u32	8	8				1			8	7	1	0
7	mul.lo.s32	8	16+7				1			23	8	1	0
8	shl.b32	8	16+7				1			0	17	0	0
9	add.s32	8	8				1			15	11	0	0
10	shl.b32	8	8				1			0	141	0	0
11	setp.lt.s32	8	8				1			0	13	0	0
12	mov.f32	8	8				1			0	32	0	0
13	@%p1 bra BB0_3;	8	8				1			0	0	0	0
14	cvta.to.global.u64	8	32				4			0	40	0	0
15	cvta.to.global.u64	8	32				4			0	35	0	0
16	mov.u32	8	8				1			0	21	0	0
17	add.s32	8	8				1			0	143	0	0
18	mov.u32	8	8				1			8	19	1	0
19	shl.b32	8	16+7				1			0	38	0	0
20	mov.u32	8	8				1			8	21	1	0
21	mad.lo.s32	8	16+7				1			0	33	0	0
22	mul.wide.s32	8	8				1			0	24	0	0
23	mov.u64	8	8				1			8	24	1	0
24	add.s64	8	16+7				1			0	26	0	0
25	mul.wide.s32	8	8				1			8	26	1	0
26	add.s64	8	16+7				1			0	37	0	0
27	mad.lo.s32	8	8				1			0	38	0	0
28	mov.u64	8	8				1			8	29	1	0
29	add.s64	8	16+7				1			23	30	1	0
30	add.s64	8	16+7				1			0	42	0	0
31	add.s64	8	8				1			0	44	0	0
32	mov.f32	8	8				1			1	46	0	0
--	BB1_2:										--	--	--
33	add.s32	8	8				1			8	34	1	0
34	mul.wide.s32	8	16+7				1			23	35	1	0
35	add.s64	8	16+7				1			23	36	1	1
36	ld.global.f32	16					4	1528		1	37	1	0
37	st.shared.f32	16				41+28	4			0	0	0	1
38	add.s32		16+7				1			23	39	1	0

39	mul.wide.s32	8	16+7				1			23	40	1	0
40	add.s64	8	16+7				1			23	41	1	1
41	ld.global.f32	16					4	1528		1	42	1	0
42	st.shared.f32	16				41+28	4			0	0	0	0
43	bar.sync 0;	8							297	1	0	0	0
44	ld.shared.f32	16				16+28	4			0	46	0	0
45	ld.shared.f32	16				32	4			32	46	1	1
46	fma.rn.f32		41+7				1			0	49	0	1
47	ld.shared.f32	16				16+28	4			0	49	0	0
48	ld.shared.f32	16				32	4			32	49	1	1
49	fma.rn.f32		41+7				1			0	52	0	1
50	ld.shared.f32	16				16+28	4			0	52	0	0
51	ld.shared.f32	16				32	4			32	52	1	1
52	fma.rn.f32		41+7				1			0	55	0	1
53	ld.shared.f32	16				16+28	4			0	55	0	0
54	ld.shared.f32	16				32	4			32	55	1	1
55	fma.rn.f32		41+7				1			0	58	0	1
56	ld.shared.f32	16				16+28	4			0	58	0	0
57	ld.shared.f32	16				32	4			32	58	1	1
58	fma.rn.f32		41+7				1			0	61	0	1
59	ld.shared.f32	16				16+28	4			0	61	0	0
60	ld.shared.f32	16				32	4			32	61	1	1
61	fma.rn.f32		41+7				1			0	64	0	1
62	ld.shared.f32	16				16+28	4			0	64	0	0
63	ld.shared.f32	16				32	4			32	64	1	1
64	fma.rn.f32		41+7				1			0	67	0	1
65	ld.shared.f32	16				16+28	4			0	67	0	0
66	ld.shared.f32	16				32	4			32	67	1	1
67	fma.rn.f32		41+7				1			0	70	0	1
68	ld.shared.f32	16				16+28	4			0	70	0	0
69	ld.shared.f32	16				32	4			32	70	1	1
70	fma.rn.f32		41+7				1			0	73	0	1
71	ld.shared.f32	16				16+28	4			0	73	0	0
72	ld.shared.f32	16				32	4			32	73	1	1
73	fma.rn.f32		41+7				1			0	76	0	1
74	ld.shared.f32	16				16+28	4			0	76	0	0
75	ld.shared.f32	16				32	4			32	76	1	1
76	fma.rn.f32		41+7				1			0	79	0	1
77	ld.shared.f32	16				16+28	4			0	79	0	0
78	ld.shared.f32	16				32	4			32	79	1	1
79	fma.rn.f32		41+7				1			0	82	0	1
80	ld.shared.f32	16				16+28	4			0	82	0	0
81	ld.shared.f32	16				32	4			32	82	1	1
82	fma.rn.f32		41+7				1			0	85	0	1
83	ld.shared.f32	16				16+28	4			0	85	0	0
84	ld.shared.f32	16				32	4			32	85	1	1
85	fma.rn.f32		41+7				1			0	88	0	1
86	ld.shared.f32	16				16+28	4			0	88	0	0
87	ld.shared.f32	16				32	4			32	88	1	1
88	fma.rn.f32		41+7				1			0	91	0	1
89	ld.shared.f32	16				16+28	4			0	91	0	0
90	ld.shared.f32	16				32	4			32	91	1	1
91	fma.rn.f32		41+7				1			0	94	0	1
92	ld.shared.f32	16				16+28	4			0	94	0	0
93	ld.shared.f32	16				32	4			32	94	1	1
94	fma.rn.f32		41+7				1			0	97	0	1
95	ld.shared.f32	16				16+28	4			0	97	0	0
96	ld.shared.f32	16				32	4			32	97	1	1
97	fma.rn.f32		41+7				1			0	100	0	1
98	ld.shared.f32	16				16+28	4			0	100	0	0
99	ld.shared.f32	16				32	4			32	100	1	1
100	fma.rn.f32		20+7				1			0	103	0	1
101	ld.shared.f32	16				16+28	4			0	103	0	0
102	ld.shared.f32	16				32	4			32	103	1	1
103	fma.rn.f32		41+7				1			0	106	0	1
104	ld.shared.f32	16				16+28	4			0	106	0	0
105	ld.shared.f32	16				32	4			32	106	1	1
106	fma.rn.f32		41+7				1			0	109	0	1
107	ld.shared.f32	16				16+28	4			0	109	0	0
108	ld.shared.f32	16				32	4			32	109	1	1
109	fma.rn.f32		41+7				1			0	112	0	1
110	ld.shared.f32	16				16+28	4			0	112	0	0
111	ld.shared.f32	16				32	4			32	112	1	1
112	fma.rn.f32		41+7				1			0	115	0	1

113	ld.shared.f32	16				16+28	4			0	115	0	0
114	ld.shared.f32	16				32	4			32	115	1	1
115	fma.rn.f32		41+7				1			0	118	0	1
116	ld.shared.f32	16				16+28	4			0	118	0	0
117	ld.shared.f32	16				32	4			32	118	1	1
118	fma.rn.f32		41+7				1			0	121	0	1
119	ld.shared.f32	16				16+28	4			0	121	0	0
120	ld.shared.f32	16				32	4			32	121	1	1
121	fma.rn.f32		41+7				1			0	124	0	1
122	ld.shared.f32	16				16+28	4			0	124	0	0
123	ld.shared.f32	16				32	4			32	124	1	1
124	fma.rn.f32		41+7				1			0	127	0	1
125	ld.shared.f32	16				16+28	4			0	127	0	0
126	ld.shared.f32	16				32	4			32	127	1	1
127	fma.rn.f32		41+7				1			0	130	0	1
128	ld.shared.f32	16				16+28	4			0	130	0	0
129	ld.shared.f32	16				32	4			32	130	1	1
130	fma.rn.f32		41+7				1			0	133	0	1
131	ld.shared.f32	16				16+28	4			0	133	0	0
132	ld.shared.f32	16				32	4			32	133	1	1
133	fma.rn.f32		41+7				1			0	136	0	1
134	ld.shared.f32	16				16+28	4			0	136	0	0
135	ld.shared.f32	16				32	4			32	136	1	1
136	fma.rn.f32		41+7				1			0	139	0	1
137	ld.shared.f32	16				16+28	4			0	139	0	0
138	ld.shared.f32	16				32	4			32	139	1	1
139	fma.rn.f32	8	41+7				1			0	155	0	0
140	bar.sync 0;								297	1	0	0	0
141	add.s32	8	16+7				1			0	0	0	0
142	add.s32	8	8				1			8	143	1	0
143	setp.le.s32	8	16+7				1			23	144	1	0
144	@%p2 bra BB1_2;	8	16+7				1			1	0	0	0
--	BB1_3:												
145	mov.u32	8	32+7				1			39	146	1	0
146	shl.b32	8	16+7				1			23	147	1	0
147	mad.lo.s32	8	16+7				1			0	149	0	0
148	mov.u32	8	8				1			8	149	1	0
149	add.s32	8	16+7				1			0	151	0	0
150	mov.u32	8	8				1			8	151	1	0
151	mad.lo.s32		16+7				1			15	153	0	1
152	cvta.to.global.u64	8	32				4			0	154	0	1
153	mul.wide.s32	8	8				1			8	154	1	0
154	add.s64	8	16+7				1			23	155	1	1
155	st.global.f32	16					4	1528		1	0	--	--
--	ret;												

Table 4-3, Level 1 and level 2 supersteps for Matrix Multiply program on GTX 760 (s_1=3, s_2=63)

Level1:	Superstep#: <start ~ end, comp, comm, ovh>, Itteration
	1: <1~32,509,0,0>, 1 2: <33~144,2947,3056,594>, 10 3: <145~155,274,1528,0>, 1
Level2:	Superstep#: <start ~ end, comp, comm, ovh>

```
1: <1~6,61,0,0>          17: <40~40,31,0,0>       33: <82~84,76,0,0>       49: <130~132,76,0,0>
2: <7~7,31,0,0>          18: <41~41,16,1528,0>    34: <85~87,76,0,0>       50: <133~135,76,0,0>
3: <8~9,39,0,0>          19: <42~43,85,0,297>     35: <88~90,76,0,0>       51: <136~138,76,0,0>
4: <10~18,128,0,0>       20: <44~45,92,0,0>       36: <91~93,76,0,0>       52: <139~140,56,0,297>
5: <19~20,39,0,0>        21: <46~48,76,0,0>       37: <94~96,76,0,0>       53: <141~142,39,0,0>
6: <21~23,47,0,0>        22: <49~51,76,0,0>       38: <97~99,76,0,0>       54: <143~143,31,0,0>
7: <24~25,39,0,0>        23: <52~54,76,0,0>       39: <100~102,76,0,0>     55: <144~144,31,0,0>
8: <26~28,47,0,0>        24: <55~57,76,0,0>       40: <103~105,76,0,0>     56: <145~145,47,0,0>
9: <29~29,31,0,0>        25: <58~60,76,0,0>       41: <106~108,76,0,0>     57: <146~146,31,0,0>
10: <30~32,47,0,0>       26: <61~63,76,0,0>       42: <109~111,76,0,0>     58: <147~148,39,0,0>
11: <33~33,16,0,0>       27: <64~66,76,0,0>       43: <112~114,76,0,0>     59: <149~150,39,0,0>
12: <34~34,31,0,0>       28: <67~69,76,0,0>       44: <115~117,76,0,0>     60: <151~152,55,0,0>
13: <35~35,31,0,0>       29: <70~72,76,0,0>       45: <118~120,76,0,0>     61: <153~153,16,0,0>
14: <36~36,16,1528,0>    30: <73~75,76,0,0>       46: <121~123,76,0,0>     62: <154~154,31,0,0>
15: <37~38,85,0,0>       31: <76~78,76,0,0>       47: <124~126,76,0,0>     63: <155~155,16,1528,0>
16: <39~39,31,0,0>       32: <79~81,76,0,0>       48: <127~129,76,0,0>
```

$parallel\_comp\ (eq.10) = 1 \times 509 + 10 \times 2947 + 1 \times 274 = 30253$

$block\_bar\_ovh\ (eq.11) = 1 \times 0 + 10 \times 594 + 1 \times 0 = 5940$

$block\_comm\ (eq.12) = 1 \times 0 + 10 \times 3056 + 1 \times 1528 = 32088$

$block\_comm_{\Delta} = 30560$

$COMP\ (eq.5) = 8 \times 10 + 30253 = 30333$

$warp\_comp = COMP \div w = 30333 \div 8 = 3792$

$warp\_comm_{\Delta} = 3820$

$$warps\_need\ (eq.15) = 4 \times \left(\left\lceil \frac{3820 \times 1122}{3792 \times (21-1)} \right\rceil + 1\right) = 232$$

$$nonoverlapped\ (eq.17) = \min\left\{\frac{32088}{8}, 191 + \frac{30560}{8} \times \max\left(0{,}1 - \frac{8 \times 4}{232}\right)\right\} = 3484$$

$block\_exec\_cycle\ (eq.4) = 553 + 5940 + 3484 + 30333 = 40310$

$comp\ (eq.19) = 36273$

$novlp\ (eq.20) = 3484$

$$\rho\ (eq.21) = max\left\{1, min\left\{\left\lfloor \frac{2048}{1024} \right\rfloor, \left\lfloor \frac{65536}{1024 \times 22} \right\rfloor, \left\lfloor \frac{48\ KB}{2048\ B} \right\rfloor\right\}\right\} = 2$$

$$\mathrm{K}\ (eq.22) = \frac{200}{6 \times 2} = 16.66$$

$$\tau\ (eq.23) = \left\lceil \frac{3484}{36273} \right\rceil + 1 = 2$$

$\mu = 3.36$

$$\rho \geq \tau : kernel\_exec\_cycle\ (eq.24) = 553 + \frac{200}{6} \times \frac{36273}{1.5} + \frac{3484}{2} = 808362$$

$mesured\_kernel\_exec\_cycle = 902152$

$$error\ \% = \frac{|902152 - 808362|}{902152} \times 100 = 10.39\ \%$$

5. Details of results for Hotspot on 940MX

$d = 3$ $n_b = 1849$ $g_0 = 19$ $\max\_thread\_per\_sm = 2048$
$n_c = 128$ $n_t = 256$ $g_1 = 160$ $reg\_per\_thread = 34$
$n_{SM} = 4$ $h = 196$ $g_2 = 134$ $shmem\_size = 48\ KB$
$n_{ws} = 4$ $l = 272$ $mem\_lat = G_2 = 313$ $shmem\_per\_block = 3072\ B$
$n_{du} = 8$ $l_c = 265$ $warp\_lnch\_ovh = 10$ $n_{reg} = 65536$
$n_{fu} = 4$ $l_m = 3$ $block\_lnch\_ovh = 382$
$warp\_size = 32$ $l_b = 4$ $issue\_cycle\ = 1$

$$w\ (eq.\ 6) = \left\lceil \frac{256}{32 \times 4} \right\rceil = 2$$

Table 5-1, Latency, FUs, Throughput, u, d and p for Hotspot on 940MX

Inst#	Instructions	FU	n^i_{fu}	T_i	n^i_{ma}	Latency		u	d	p
						Comp.	Comm.			
1	ld.param.u32 %r9, [_Kerrnel_param_0];	SPs	32	32		6		15	0	0
2	ld.param.u64 %rd3, [_Kerrnel_param_1];	SPs	32	32		6		49	0	0
3	ld.param.u64 %rd4, [_Kerrnel_param_2];	SPs	32	32		6		39	0	0
4	ld.param.u64 %rd5, [_Kerrnel_param_3];	SPs	32	32		6		193	0	0
5	ld.param.u32 %r10, [_Kerrnel_param_4];	SPs	32	32		6		32	0	0
6	ld.param.u32 %r11, [_Kerrnel_param_5];	SPs	32	32		6		28	0	0
7	ld.param.u32 %r12, [_Kerrnel_param_6];	SPs	32	32		6		22	0	0
8	ld.param.u32 %r13, [_Kerrnel_param_7];	SPs	32	32		6		20	0	0
9	ld.param.f32 %f2, [_Kerrnel_param_8];	SPs	32	32		6		59	0	0
10	ld.param.f32 %f3, [_Kerrnel_param_9];	SPs	32	32		6		60	0	0
11	ld.param.f32 %f4, [_Kerrnel_param_10];	SPs	32	32		6		61	0	0
12	ld.param.f32 %f5, [_Kerrnel_param_11];	SPs	32	32		6		62	0	0
13	ld.param.f32 %f6, [_Kerrnel_param_12];	SPs	32	32		6		59	0	0
14	mov.u32 %r14, %ctaid.x;	SPs	32	32		27		21	0	0
15	shl.b32 %r15, %r9, 1;	SPs	32	16		6		17	0	0
16	mov.u32 %r16, 16;	SPs	32	32		6		17	1	0
17	sub.s32 %r17, %r16, %r15;	SPs	32	32		6		19	0	0
18	mov.u32 %r18, %ctaid.y;	SPs	32	32		27		19	1	0
19	mul.lo.s32 %r19, %r18, %r17;	SPs	32	32		6		20	1	0
20	sub.s32 %r20, %r19, %r13;	SPs	32	32		6		24	0	0
21	mul.lo.s32 %r21, %r14, %r17;	SPs	32	32		6		22	1	0
22	sub.s32 %r22, %r21, %r12;	SPs	32	32		6		26	0	0
23	mov.u32 %r23, %tid.y;	SPs	32	32		27		24	1	0
24	add.s32 %r1, %r20, %r23;	SPs	32	32		6		27	0	0
25	mov.u32 %r24, %tid.x;	SPs	32	32		27		26	1	0
26	add.s32 %r2, %r22, %r24;	SPs	32	32		6		33	0	0
27	setp.gt.s32 %p1, %r1, -1;	SPs	32	16		6		30	0	0
28	add.s32 %r25, %r11, -1;	SPs	32	32		6		29	1	0
29	setp.le.s32 %p2, %r1, %r25;	SPs	32	16		6		30	1	0
30	and.pred %p3, %p1, %p2;	SPs	32	32		6		35	0	0
31	setp.gt.s32 %p4, %r2, -1;	SPs	32	16		6		34	0	0
32	add.s32 %r26, %r10, -1;	SPs	32	32		6		33	1	0
33	setp.le.s32 %p5, %r2, %r26;	SPs	32	16		6		34	1	0
34	and.pred %p6, %p4, %p5;	SPs	32	32		6		35	1	0
35	and.pred %p7, %p3, %p6;	SPs	32	32		6		36	1	0
36	@!%p7 bra BB0_2;	SPs	32	16		6		0	0	0
37	bra.uni BB0_1;	SPs	32	32		6		0	0	0
--	BB0_1:							--	--	--
38	mad.lo.s32 %r27, %r1, %r10, %r2;	SPs	32	32		6		40	0	0
39	cvta.to.global.u64 %rd6, %rd4;	SPs	32	8		6		41	0	0
40	mul.wide.s32 %rd7, %r27, 4;	SPs	32	32		6		41	1	0
41	add.s64 %rd8, %rd6, %rd7;	SPs	32	32		6		42	1	1
42	ld.global.f32 %f7, [%rd8];	LDST	8	8	1~2		313	48	0	1
43	mul.wide.s32 %rd9, %r23, 64;	SPs	32	32		6		45	0	0
44	mov.u64 %rd10, Kerrnel$__cuda_...;	SPs	32	32		6		45	1	0
45	add.s64 %rd11, %rd10, %rd9;	SPs	32	32		6		47	0	0
46	mul.wide.s32 %rd12, %r24, 4;	SPs	32	32		6		47	1	0
47	add.s64 %rd13, %rd11, %rd12;	SPs	32	32		6		48	1	1
48	st.shared.f32 [%rd13], %f7;	LDST	8	8	1	20		0	0	1
49	cvta.to.global.u64 %rd14, %rd3;	SPs	32	8		6		50	1	0
50	add.s64 %rd15, %rd14, %rd7;	SPs	32	32		6		51	1	1
51	ld.global.f32 %f8, [%rd15];	LDST	8	8	1~2		313	55	0	1
52	mov.u64 %rd16, _Kerrnel$__cuda_...;	SPs	32	32		6		53	1	0
53	add.s64 %rd17, %rd16, %rd9;	SPs	32	32		6		54	1	0
54	add.s64 %rd18, %rd17, %rd12;	SPs	32	32		6		55	1	1

55	st.shared.f32 [%rd18], %f8;	LDST	8	8	1	20		0	0	0
--	BB0_2:							--	--	--
56	bar.sync 0;		MI	--		120		0	0	0
57	setp.lt.s32 %p8, %r9, 1;	SPs	32	16		6		58	1	0
58	@%p8 bra BB0_11;	SPs	32	16		6		0	0	0
59	div.rn.f32 %f9, %f6, %f2;	SFU	8	8		137		63	0	0
60	rcp.rn.f32 %f10, %f3;	SFU	8	8		370		93	0	0
61	rcp.rn.f32 %f11, %f4;	SFU	8	8		370		79	0	0
62	rcp.rn.f32 %f1, %f5;	SFU	8	8		370		158	0	1
63	cvt.f64.f32 %fd1, %f9;	SPs	32	1		6		161	0	0
64	add.s32 %r37, %r20, 15;	SPs	32	32		6		65	1	0
65	setp.gt.s32 %p9, %r37, %r25;	SPs	32	16		6		70	0	0
66	mov.u32 %r39, -15;	SPs	32	32		6		67	1	0
67	sub.s32 %r40, %r39, %r20;	SPs	32	32		6		68	1	0
68	add.s32 %r41, %r11, %r40;	SPs	32	32		6		69	1	0
69	add.s32 %r42, %r41, 14;	SPs	32	32		6		70	1	0
70	selp.b32%r43, %r42, 15, %p9;	SPs	32	16		6		71	1	0
71	setp.lt.s32 %p10, %r23, %r43;	SPs	32	16		6		73	0	0
72	add.s32 %r45, %r23, 1;	SPs	32	32		6		73	1	0
73	selp.b32%r46, %r45, %r43, %p10;	SPs	32	16		6		74	1	0
74	mul.wide.s32 %rd19, %r46, 64;	SPs	32	32		6		76	0	0
75	mov.u64 %rd20, _Kerrnel$__cuda_...;	SPs	32	32		6		76	1	0
76	add.s64 %rd21, %rd20, %rd19;	SPs	32	32		6		78	0	0
77	mul.wide.s32 %rd22, %r24, 4;	SPs	32	32		6		78	1	0
78	add.s64 %rd1, %rd21, %rd22;	SPs	32	32		6		139	0	0
79	cvt.f64.f32 %fd2, %f11;	SPs	32	1		6		144	0	0
80	add.s32 %r51, %r22, 15;	SPs	32	32		6		81	1	0
81	setp.gt.s32 %p11, %r51, %r26;	SPs	32	16		6		85	0	0
82	sub.s32 %r53, %r39, %r22;	SPs	32	32		6		83	1	0
83	add.s32 %r54, %r10, %r53;	SPs	32	32		6		84	1	0
84	add.s32 %r55, %r54, 14;	SPs	32	32		6		85	1	0
85	selp.b32%r56, %r55, 15, %p11;	SPs	32	16		6		86	1	0
86	setp.lt.s32 %p12, %r24, %r56;	SPs	32	16		6		88	0	0
87	add.s32 %r57, %r24, 1;	SPs	32	32		6		88	1	0
88	selp.b32%r58, %r57, %r56, %p12;	SPs	32	16		6		89	1	0
89	mul.wide.s32 %rd23, %r58, 4;	SPs	32	32		6		92	0	0
90	mul.wide.s32 %rd24, %r23, 64;	SPs	32	32		6		91	1	0
91	add.s64 %rd25, %rd20, %rd24;	SPs	32	32		6		92	1	0
92	add.s64 %rd2, %rd25, %rd23;	SPs	32	32		6		151	0	0
93	cvt.f64.f32 %fd3, %f10;	SPs	32	1		6		155	0	0
94	mov.u32 %r59, 1;	SPs	32	32		6		95	1	0
95	sub.s32 %r129, %r59, %r9;	SPs	32	32		6		169	0	0
96	mov.u32 %r130, 0;	SPs	32	32		6		98	0	0
--	BB0_4:							--	--	--
97	mov.u32 %r61, 14;	SPs	32	32		6		98	1	0
98	sub.s32 %r6, %r61, %r130;	SPs	32	32		6		99	1	0
99	setp.le.s32 %p13, %r24, %r6;	SPs	32	16		6		102	0	0
100	add.s32 %r130, %r130, 1;	SPs	32	32		6		101	1	0
101	setp.ge.s32 %p14, %r24, %r130;	SPs	32	16		6		102	1	0
102	and.pred %p15, %p13, %p14;	SPs	32	32		6		104	0	0
103	mov.u16 %rs8, 0;	SPs	32	32		6		171	0	0
104	@!%p15 bra BB0_7;	SPs	32	16		6		0	0	0
105	bra.uni BB0_5;	SPs	32	32		6		0	0	0
--	BB0_5:							--	--	--
106	setp.gt.s32 %p16, %r23, %r6;	SPs	32	16		6		108	0	0
107	setp.lt.s32 %p17, %r23, %r130;	SPs	32	16		6		108	1	0
108	or.pred %p18, %p17, %p16;	SPs	32	32		6		115	0	0
109	shr.s32 %r69, %r22, 31;	SPs	32	16		6		112	0	0
110	setp.gt.s32 %p20, %r24, %r56;	SPs	32	16		6		114	0	0
111	neg.s32 %r78, %r22;	SPs	32	16		6		112	1	0
112	and.b32 %r79, %r69, %r78;	SPs	32	32		6		113	1	0
113	setp.lt.s32 %p21, %r24, %r79;	SPs	32	16		6		114	1	0
114	or.pred %p22, %p21, %p20;	SPs	32	32		6		115	1	0
115	or.pred %p23, %p18, %p22;	SPs	32	32		6		122	0	0
116	shr.s32 %r83, %r20, 31;	SPs	32	16		6		119	0	0
117	setp.gt.s32 %p25, %r23, %r43;	SPs	32	16		6		121	0	0
118	neg.s32 %r90, %r20;	SPs	32	16		6		119	1	0
119	and.b32 %r91, %r83, %r90;	SPs	32	32		6		120	1	0
120	setp.lt.s32 %p26, %r23, %r91;	SPs	32	16		6		121	1	0
121	or.pred %p27, %p26, %p25;	SPs	32	32		6		122	1	0
122	or.pred %p28, %p23, %p27;	SPs	32	32		6		123	1	0
123	@%p28 bra BB0_7;	SPs	32	16		6		0	0	0
124	add.s64 %rd30, %rd25, %rd22;	SPs	32	32		6		125	1	1
125	ld.shared.f32 %f12, [%rd30];	LDST	8	8	1	6		126	1	1

126	cvt.f64.f32 %fd4, %f12;	SPs	32	1		6		142	0	0
127	mov.u64 %rd31, Kerrnel$__cuda_...;	SPs	32	32		6		128	1	0
128	add.s64 %rd32, %rd31, %rd24;	SPs	32	32		6		129	1	0
129	add.s64 %rd33, %rd32, %rd22;	SPs	32	32		6		130	1	1
130	ld.shared.f32 %f13, [%rd33];	LDST	8	8	1	6		131	1	1
131	cvt.f64.f32 %fd5, %f13;	SPs	32	1		6		144	0	0
132	setp.gt.s32 %p29, %r23, %r91;	SPs	32	16		6		134	0	0
133	add.s32 %r103, %r23, -1;	SPs	32	32		6		134	1	0
134	selp.b32 %r104, %r103, %r91, %p29;	SPs	32	16		6		135	1	0
135	mul.wide.s32 %rd34, %r104, 64;	SPs	32	32		6		136	1	0
136	add.s64 %rd35, %rd20, %rd34;	SPs	32	32		6		137	1	0
137	add.s64 %rd36, %rd35, %rd22;	SPs	32	32		6		138	1	1
138	ld.shared.f32 %f14, [%rd36];	LDST	8	8	1	6		140	0	0
139	ld.shared.f32 %f15, [%rd1];	LDST	8	8	1	6		140	1	1
140	add.f32 %f16, %f15, %f14;	SPs	32	32		6		141	1	0
141	cvt.f64.f32 %fd6, %f16;	SPs	32	1		6		143	0	1
142	add.f64 %fd7, %fd4, %fd4;	DPU	1	1		6		143	1	0
143	sub.f64 %fd8, %fd6, %fd7;	DPU	1	1		6		144	1	0
144	fma.rn.f64 %fd9, %fd2, %fd8, %fd5;	DPU	1	1		65		155	0	1
145	setp.gt.s32 %p30, %r24, %r79;	SPs	32	16		6		147	0	0
146	add.s32 %r111, %r24, -1;	SPs	32	32		6		147	1	0
147	selp.b32 %r112, %r111, %r79, %p30;	SPs	32	16		6		148	1	0
148	mul.wide.s32 %rd37, %r112, 4;	SPs	32	32		6		149	1	0
149	add.s64 %rd38, %rd25, %rd37;	SPs	32	32		6		150	1	1
150	ld.shared.f32 %f17, [%rd38];	LDST	8	8	1	6		152	0	0
151	ld.shared.f32 %f18, [%rd2];	LDST	8	8	1	6		152	1	1
152	add.f32 %f19, %f18, %f17;	SPs	32	32		6		153	1	0
153	cvt.f64.f32 %fd10, %f19;	SPs	32	1		6		154	1	0
154	sub.f64 %fd11, %fd10, %fd7;	DPU	1	1		6		155	1	0
155	fma.rn.f64 %fd12, %fd3, %fd11, %fd9;	DPU	1	1		65		160	0	1
156	mov.f32 %f20, 0f42A00000;	SPs	32	32		6		157	1	0
157	sub.f32 %f21, %f20, %f12;	SPs	32	32		6		158	1	0
158	mul.f32 %f22, %f1, %f21;	SPs	32	32		6		159	1	0
159	cvt.f64.f32 %fd13, %f22;	SPs	32	1		6		160	1	0
160	add.f64 %fd14, %fd13, %fd12;	DPU	1	1		6		161	1	0
161	fma.rn.f64 %fd15, %fd1, %fd14, %fd4;	DPU	1	1		65		162	1	1
162	cvt.rn.f32.f64 %f23, %fd15;	SPs	32	8		6		166	0	0
163	mov.u64 %rd39, _Kerrnel$__cuda_...;	SPs	32	32		6		164	1	0
164	add.s64 %rd40, %rd39, %rd24;	SPs	32	32		6		165	1	0
165	add.s64 %rd41, %rd40, %rd22;	SPs	32	32		6		166	1	1
166	st.shared.f32 [%rd41], %f23;	LDST	8	8	1	20		0	0	1
167	mov.u16 %rs8, 1;	SPs	32	32		6		171	0	0
--	BB0_7:							--	--	--
168	bar.sync 0;		MI			120		0	0	0
169	setp.eq.s32 %p31, %r129, 0;	SPs	32	16		6		170	1	0
170	@%p31 bra BB0_11;	SPs	32	16		6		0	0	0
171	setp.eq.s16 %p32, %rs8, 0;	SPs	32	16		6		172	1	0
172	@%p32 bra BB0_10;	SPs	32	16		6		0	0	0
173	mov.u64 %rd43, _Kerrnel$__cuda_...;	SPs	32	32		6		174	1	0
174	add.s64 %rd44, %rd43, %rd24;	SPs	32	32		6		175	1	0
175	add.s64 %rd46, %rd44, %rd22;	SPs	32	32		6		176	1	1
176	ld.shared.f32 %f24, [%rd46];	LDST	8	8	1	6		178	0	1
177	add.s64 %rd49, %rd25, %rd22;	SPs	32	32		6		178	1	1
178	st.shared.f32 [%rd49], %f24;	LDST	8	8	1	20		0	0	0
--	BB0_10:							--	--	--
179	bar.sync 0;		MI			120		0	0	0
180	add.s32 %r129, %r129, 1;	SPs	32	32		6		0	0	0
181	setp.lt.s32 %p33, %r130, %r9;	SPs	32	16		6		182	1	0
182	@%p33 bra BB0_4;	SPs	32	16		6		0	0	0
--	BB0_11:							--	--	--
183	and.b16 %rs7, %rs8, 255;	SPs	32	32		6		184	1	0
184	setp.eq.s16 %p34, %rs7, 0;	SPs	32	16		6		185	1	0
185	@%p34 bra BB0_13;	SPs	32	16		6		0	0	0
186	mul.wide.s32 %rd50, %r23, 64;	SPs	32	32		6		188	0	0
187	mov.u64 %rd51, _Kerrnel$__cuda_...;	SPs	32	32		6		188	1	0
188	add.s64 %rd52, %rd51, %rd50;	SPs	32	32		6		190	0	0
189	mul.wide.s32 %rd53, %r24, 4;	SPs	32	32		6		190	1	0
190	add.s64 %rd54, %rd52, %rd53;	SPs	32	32		6		191	1	1
191	ld.shared.f32 %f25, [%rd54];	LDST	8	8	1	6		196	0	1
192	mad.lo.s32 %r128, %r1, %r10, %r2;	SPs	32	32		6		194	0	0
193	cvta.to.global.u64 %rd55, %rd5;	SPs	32	8		6		195	0	0
194	mul.wide.s32 %rd56, %r128, 4;	SPs	32	32		6		195	1	0
195	add.s64 %rd57, %rd55, %rd56;	SPs	32	32		6		196	1	1
196	st.global.f32 [%rd57], %f25;	LDST	8	8	1~2		313	0	0	0

--	BB0_13:									
--	ret;									

Table 5-2, BLOCK data structure for Hotspot on 940MX

Inst#	Instructions	$comp_i$						$comm_i$	ovh_i	$sync_i$	u	d	p
		Issue	FU_1	FU_2	FU_3	FU_4	$32^1/T_i$						
1	ld.param.u32	2	6+1				1			0	15	0	0
2	ld.param.u64	2	2				1			0	49	0	0
3	ld.param.u64	2	2				1			0	39	0	0
4	ld.param.u64	2	2				1			0	193	0	0
5	ld.param.u32	2	2				1			0	32	0	0
6	ld.param.u32	2	2				1			0	28	0	0
7	ld.param.u32	2	2				1			0	22	0	0
8	ld.param.u32	2	2				1			0	20	0	0
9	ld.param.f32	2	2				1			0	59	0	0
10	ld.param.f32	2	2				1			0	60	0	0
11	ld.param.f32	2	2				1			0	61	0	0
12	ld.param.f32	2	2				1			0	62	0	0
13	ld.param.f32	2	2				1			0	59	0	0
14	mov.u32	2	2				1			0	21	0	0
15	shl.b32	2	4				2			0	17	0	0
16	mov.u32	2	2				1			2	17	1	0
17	sub.s32	2	6+1				1			0	19	0	0
18	mov.u32	2	2				1			2	19	1	0
19	mul.lo.s32	2	6+1				1			7	20	1	0
20	sub.s32	2	6+1				1			0	24	0	0
21	mul.lo.s32	2	2				1			2	22	1	0
22	sub.s32	2	6+1				1			0	26	0	0
23	mov.u32	2	2				1			2	24	1	0
24	add.s32	2	6+1				1			0	27	0	0
25	mov.u32	2	2				1			2	26	1	0
26	add.s32	2	6+1				1			0	33	0	0
27	setp.gt.s32	2	4				2			0	30	0	0
28	add.s32	2	2				1			2	29	1	0
29	setp.le.s32	2	6+2				2			8	30	1	0
30	and.pred	2	6+1				1			0	35	0	0
31	setp.gt.s32	2	4				2			0	34	0	0
32	add.s32	2	2				1			2	33	1	0
33	setp.le.s32	2	6+2				2			8	34	1	0
34	and.pred	2	6+1				1			7	35	1	0
35	and.pred	2	6+1				1			7	36	1	0
36	@!%p7 bra BB0_2;	2	6+2				2			1	0	0	0
37	bra.uni BB0_1;	2	2				1			0	0	0	0
--	BB0_1:										--	--	--
38	mad.lo.s32	2	6+1				1			5	40	0	0
39	cvta.to.global.u64	2	8				4			0	41	0	0
40	mul.wide.s32	2	2				1			7	41	1	0
41	add.s64	2	6+1				1			7	42	1	1
42	ld.global.f32	8					--	1252		0	48	0	1
43	mul.wide.s32		2				1			1	45	0	0
44	mov.u64	2	2				1			2	45	1	0
45	add.s64	2	6+1				1			0	47	0	0
46	mul.wide.s32	2	2				1			2	47	1	0
47	add.s64	2	6+1				1			7	48	1	1
48	st.shared.f32	8				20+4	4			0	0	0	1
49	cvta.to.global.u64		8				4			8	50	1	0
50	add.s64	2	6+1				1			7	51	1	1
51	ld.global.f32	8					--	1252		0	55	0	1
52	mov.u64		2				1			1	53	1	0
53	add.s64	2	6+1				1			7	54	1	0
54	add.s64	2	6+1				1			7	55	1	1
55	st.shared.f32	8				20+4	4			0	0	0	0
--	BB0_2:										--	--	--
56	bar.sync	2					--		120	1	0	0	0
57	setp.lt.s32	2	6+2				2			8	58	1	0
58	@%p8 bra BB0_11;	2	6+2				2			1	0	0	0
59	div.rn.f32	8			137+4		4			117	63	0	0
60	rcp.rn.f32	8			8		4			0	93	0	0
61	rcp.rn.f32	8			8		4			0	79	0	0
62	rcp.rn.f32	8			8		4			0	158	0	1

[1] warp_size

63	cvt.f64.f32		2				1			0	161	0	0
64	add.s32	2	2				1			2	65	1	0
65	setp.gt.s32	2	6+2				2			0	70	0	0
66	mov.u32	2	2				1			2	67	1	0
67	sub.s32	2	6+1				1			7	68	1	0
68	add.s32	2	6+1				1			7	69	1	0
69	add.s32	2	6+1				1			7	70	1	0
70	selp.b32	2	6+2				2			8	71	1	0
71	setp.lt.s32	2	4				2			0	73	0	0
72	add.s32	2	2				1			2	73	1	0
73	selp.b32	2	6+2				2			8	74	1	0
74	mul.wide.s32	2	6+1				1			0	76	0	0
75	mov.u64	2	2				1			2	76	1	0
76	add.s64	2	6+1				1			0	78	0	0
77	mul.wide.s32	2	2				1			2	78	1	0
78	add.s64	2	6+1				1			0	139	0	0
79	cvt.f64.f32	2	12				32			0	144	0	0
80	add.s32	2	2				1			2	81	1	0
81	setp.gt.s32	2	6+2				2			0	85	0	0
82	sub.s32	2	2				1			2	83	1	0
83	add.s32	2	6+1				1			7	84	1	0
84	add.s32	2	6+1				1			7	85	1	0
85	selp.b32	2	6+2				2			8	86	1	0
86	setp.lt.s32	2	6+2				2			0	88	0	0
87	add.s32	2	2				1			2	88	1	0
88	selp.b32	2	6+2				2			8	89	1	0
89	mul.wide.s32	2	6+1				1			0	92	0	0
90	mul.wide.s32	2	2				1			2	91	1	0
91	add.s64	2	6+1				1			7	92	1	0
92	add.s64	2	6+1				1			0	151	0	0
93	cvt.f64.f32	2	12				32			0	155	0	0
94	mov.u32	2	2				1			2	95	1	0
95	sub.s32	2	6+1				1			0	169	0	0
96	mov.u32	2	2				1			1	98	0	0
--	BB0_4:									--	--	--	--
97	mov.u32	2	6+1				1			7	98	1	0
98	sub.s32	2	6+1				1			7	99	1	0
99	setp.le.s32	2	6+2				2			0	102	0	0
100	add.s32	2	2				1			2	101	1	0
101	setp.ge.s32	2	6+2				2			8	102	1	0
102	and.pred	2	6+1				1			0	104	0	0
103	mov.u16	2	2				1			0	171	0	0
104	@!%p15 bra BB0_7;	2	4				2			1	0	0	0
105	bra.uni BB0_5;	2	2				1			0	0	0	0
--	BB0_5:									--	--	--	--
106	setp.gt.s32	2	4				2			0	108	0	0
107	setp.lt.s32	2	4				2			4	108	1	0
108	or.pred	2	6+1				1			0	115	0	0
109	shr.s32	2	4				2			0	112	0	0
110	setp.gt.s32	2	4				2			0	114	0	0
111	neg.s32	2	4				2			4	112	1	0
112	and.b32	2	6+1				1			7	113	1	0
113	setp.lt.s32	2	6+2				2			8	114	1	0
114	or.pred	2	6+1				1			7	115	1	0
115	or.pred	2	6+1				1			0	122	0	0
116	shr.s32	2	4				2			0	119	0	0
117	setp.gt.s32	2	4				2			0	121	0	0
118	neg.s32	2	4				2			4	119	1	0
119	and.b32	2	6+1				1			7	120	1	0
120	setp.lt.s32	2	6+2				2			8	121	1	0
121	or.pred	2	6+1				1			7	122	1	0
122	or.pred	2	6+1				1			7	123	1	0
123	@%p28 bra BB0_7;	2	6+2				2			1	0	0	0
124	add.s64	2	2				1			2	125	1	1
125	ld.shared.f32	8				6+4	4			10	126	1	1
126	cvt.f64.f32	2	12				32			0	142	0	0
127	mov.u64	2	2				1			2	128	1	0
128	add.s64	2	6+1				1			7	129	1	0
129	add.s64	2	6+1				1			7	130	1	1
130	ld.shared.f32	8				6+4	4			10	131	1	1
131	cvt.f64.f32	2	12				32			0	144	0	0
132	setp.gt.s32	2	4				2			0	134	0	0
133	add.s32	2	2				1			2	134	1	0
134	selp.b32	2	6+2				2			8	135	1	0

135	mul.wide.s32	2	6+1				1			7	136	1	0
136	add.s64	2	6+1				1			7	137	1	0
137	add.s64	2	6+1				1			7	138	1	1
138	ld.shared.f32	8				6+4	4			0	140	0	0
139	ld.shared.f32	8				8	4			8	140	1	1
140	add.f32	2	6+1				1			7	141	1	0
141	cvt.f64.f32		12				32			0	143	0	1
142	add.f64	64		12			32			12	143	1	0
143	sub.f64	64		12			32			12	144	1	0
144	fma.rn.f64	64		65+32			32			0	155	0	1
145	setp.gt.s32		4				2			0	147	0	0
146	add.s32	2	2				1			2	147	1	0
147	selp.b32	2	6+2				2			8	148	1	0
148	mul.wide.s32	2	6+1				1			7	149	1	0
149	add.s64	2	6+1				1			7	150	1	1
150	ld.shared.f32	8				6+4	4			0	152	0	0
151	ld.shared.f32	8				8	4			8	152	1	1
152	add.f32	2	6+1				1			7	153	1	0
153	cvt.f64.f32	2	12				32			12	154	1	0
154	sub.f64	64		12			32			12	155	1	0
155	fma.rn.f64	64		65+32			32			0	160	0	1
156	mov.f32		2				1			2	157	1	0
157	sub.f32	2	6+1				1			7	158	1	0
158	mul.f32	2	6+1				1			7	159	1	0
159	cvt.f64.f32	2	12				32			12	160	1	0
160	add.f64	64		12			32			12	161	1	0
161	fma.rn.f64	64		65+32			32			97	162	1	1
162	cvt.rn.f32.f64	2	6+4				4			0	166	0	0
163	mov.u64	2	2				1			2	164	1	0
164	add.s64	2	6+1				1			7	165	1	0
165	add.s64	2	6+1				1			7	166	1	1
166	st.shared.f32	8				20+4	4			0	0	0	1
167	mov.u16		2				1			0	171	0	0
--	BB0_7:									--	--	--	--
168	bar.sync	2							120	1	0	0	0
169	setp.eq.s32	2	6+2				2			8	170	1	0
170	@%p31 bra BB0_11;	2	6+2				2			1	0	0	0
171	setp.eq.s16	2	6+2				2			8	172	1	0
172	@%p32 bra BB0_10;	2	6+2				2			1	0	0	0
173	mov.u64	2	2				1			2	174	1	0
174	add.s64	2	6+1				1			7	175	1	0
175	add.s64	2	6+1				1			7	176	1	1
176	ld.shared.f32	8				6+4	4			0	178	0	1
177	add.s64		2				1			2	178	1	1
178	st.shared.f32	8				20+4	4			0	0	0	0
--	BB0_10:									--	--	--	--
179	bar.sync	2							120	1	0	0	0
180	add.s32	2	6+1				1			0	0	0	0
181	setp.lt.s32	2	4				2			4	182	1	0
182	@%p33 bra BB0_4;	2	6+2				2			1	0	0	0
--	BB0_11:									--	--	--	--
183	and.b16	2	2				1			2	184	1	0
184	setp.eq.s16	2	6+2				2			8	185	1	0
185	@%p34 bra BB0_13;	2	6+2				2			1	0	0	0
186	mul.wide.s32	2	2				1			0	188	0	0
187	mov.u64	2	2				1			2	188	1	0
188	add.s64	2	6+1				1			0	190	0	0
189	mul.wide.s32	2	2				1			2	190	1	0
190	add.s64	2	6+1				1			7	191	1	1
191	ld.shared.f32	8				6+4	4			0	196	0	1
192	mad.lo.s32		2				1			0	194	0	0
193	cvta.to.global.u64	2	8				4			0	195	0	0
194	mul.wide.s32	2	2				1			2	195	1	0
195	add.s64	2	6+1				1			7	196	1	1
196	st.global.f32	8					--	1252		1	0	0	0
--	BB0_13:												
--	ret;												

Table 5-3, Level 1 and Level 2 supersteps for Hotspot on 940MX (s_1=4, s_2=119)

Level1:	Superstep#: <start ~ end, comp, comm, ovh>, Itteration
	1: <1~96,773,2504,120>, 1 2: <97~170,1318,0,120>, 2
	3: <171~182,115,0,120>, 1 4: <183~196,87,1252,0>, 1

Level2: Superstep#: <start ~ end, comp, comm, ovh>

1: <1~16,41,0,0>	31: <65~66,12,0,0>	61: <115~118,21,0,0>	91: <157~157,9,0,0>
2: <17~18,11,0,0>	32: <67~67,9,0,0>	62: <119~119,9,0,0>	92: <158~158,9,0,0>
3: <19~19,9,0,0>	33: <68~68,9,0,0>	63: <120~120,10,0,0>	93: <159~159,14,0,0>
4: <20~21,11,0,0>	34: <69~69,9,0,0>	64: <121~121,9,0,0>	94: <160~160,76,0,0>
5: <22~23,11,0,0>	35: <70~70,10,0,0>	65: <122~122,9,0,0>	95: <161~161,161,0,0>
6: <24~25,11,0,0>	36: <71~72,8,0,0>	66: <123~123,10,0,0>	96: <162~163,14,0,0>
7: <26~28,15,0,0>	37: <73~73,10,0,0>	67: <124~124,4,0,0>	97: <164~164,9,0,0>
8: <29~29,10,0,0>	38: <74~75,11,0,0>	68: <125~125,18,0,0>	98: <165~165,9,0,0>
9: <30~32,15,0,0>	39: <76~77,11,0,0>	69: <126~127,16,0,0>	99: <166~168,32,0,120>
10: <33~33,10,0,0>	40: <78~80,23,0,0>	70: <128~128,9,0,0>	100: <169~169,10,0,0>
11: <34~34,9,0,0>	41: <81~82,12,0,0>	71: <129~129,9,0,0>	101: <170~170,10,0,0>
12: <35~35,9,0,0>	42: <83~83,9,0,0>	72: <130~130,18,0,0>	102: <171~171,10,0,0>
13: <36~36,10,0,0>	43: <84~84,9,0,0>	73: <131~133,20,0,0>	103: <172~172,10,0,0>
14: <37~38,11,0,0>	44: <85~85,10,0,0>	74: <134~134,10,0,0>	104: <173~173,4,0,0>
15: <39~40,17,0,0>	45: <86~87,12,0,0>	75: <135~135,9,0,0>	105: <174~174,9,0,0>
16: <41~41,9,0,0>	46: <88~88,10,0,0>	76: <136~136,9,0,0>	106: <175~175,9,0,0>
17: <42~43,10,1252,0>	47: <89~90,11,0,0>	77: <137~137,9,0,0>	107: <176~177,18,0,0>
18: <44~44,4,0,0>	48: <91~91,9,0,0>	78: <138~139,26,0,0>	108: <178~179,32,0,120>
19: <45~46,11,0,0>	49: <92~94,23,0,0>	79: <140~140,9,0,0>	109: <180~181,13,0,0>
20: <47~47,9,0,0>	50: <95~96,11,0,0>	80: <141~142,64,0,0>	110: <182~182,10,0,0>
21: <48~49,32,0,0>	51: <97~97,9,0,0>	81: <143~143,76,0,0>	111: <183~183,4,0,0>
22: <50~50,9,0,0>	52: <98~98,9,0,0>	82: <144~146,161,0,0>	112: <184~184,10,0,0>
23: <51~52,10,1252,0>	53: <99~100,12,0,0>	83: <147~147,10,0,0>	113: <185~185,10,0,0>
24: <53~53,9,0,0>	54: <101~101,10,0,0>	84: <148~148,9,0,0>	114: <186~187,6,0,0>
25: <54~54,9,0,0>	55: <102~104,15,0,0>	85: <149~149,9,0,0>	115: <188~189,11,0,0>
26: <55~56,32,0,120>	56: <105~107,12,0,0>	86: <150~151,26,0,0>	116: <190~190,9,0,0>
27: <57~57,10,0,0>	57: <108~111,21,0,0>	87: <152~152,9,0,0>	117: <191~194,20,0,0>
28: <58~58,10,0,0>	58: <112~112,9,0,0>	88: <153~153,14,0,0>	118: <195~195,9,0,0>
29: <59~59,149,0,0>	59: <113~113,10,0,0>	89: <154~154,76,0,0>	119: <196~196,8,1252,0>
30: <60~64,32,0,0>	60: <114~114,9,0,0>	90: <155~156,161,0,0>	

$parallel\_comp\ (eq.10) = 1 \times 773 + 2 \times 1318 + 1 \times 115 + 1 \times 87 = 3611$

$block\_bar\_ovh\ (eq.11) = 1 \times 120 + 2 \times 120 + 1 \times 120 + 1 \times 0 = 480$

$block\_comm\ (eq.12) = 1 \times 2504 + 2 \times 0 + 1 \times 0 + 1 \times 1252 = 3756$

$block\_comm_{\Delta} = 2504$

$COMP\ (eq.5) = 2 \times 10 + 3611 = 3631$

$warp\_comp = COMP \div w = 3631 \div 2 = 1806$

$warp\_comm_{\Delta} = 1252$

$$warps\_need\ (eq.15) = 4 \times \left(\left\lceil \frac{1252 \times 265}{1806 \times (3-1)} \right\rceil + 1\right) = 372$$

$$nonoverlapped\ (eq.17) = \min\left\{\frac{3756}{2}, 313 + \frac{2504}{2} \times \max\left(0{,}1 - \frac{2 \times 4}{372}\right)\right\} = 1538$$

$block\_exec\_cycle\ (eq.4) = 382 + 480 + 1538 + 3631 = 6031$

$comp\ (eq.19) = 4111$

$novlp\ (eq.20) = 1538$

$$\rho\ (eq.21) = max\left\{1, min\left\{\left\lfloor \frac{2048}{256} \right\rfloor, \left\lfloor \frac{65536}{256 \times 34} \right\rfloor, \left\lfloor \frac{48\ KB}{3072\ B} \right\rfloor\right\}\right\} = 7$$

$$\mathrm{K}\ (eq.22) = \frac{1849}{4 \times 7} = 66.035$$

$$\tau\ (eq.23) = \left\lceil \frac{1538}{4111} \right\rceil + 1 = 2$$

$\mu = 1.93$

$$\rho \geq \tau : kernel\_exec\_cycle\ (eq.24) = 382 + \frac{1849}{4} \times \frac{4111}{1.93} + \frac{1538}{2} = 985768$$

$mesured\_kernel\_exec\_cycle = 1044800$

$$error\ \% = \frac{|1044800 - 985768|}{1044800} \times 100 = 5.65\ \%$$

6. Details of results for KNN on 940MX

$d = 3$	$n_b = 168$	$g_0 = 19$	$\max\_thread\_per\_sm = 2048$
$n_c = 128$	$n_t = 256$	$g_1 = 160$	$reg\_per\_thread = 9$
$n_{SM} = 4$	$h = 28$	$g_2 = 134$	$shmem\_size = 48\ KB$
$n_{ws} = 4$	$l = 28$	$mem\_lat = G_2 = 313$	$shmem\_per\_block = 0\ B$
$n_{du} = 8$	$l_c = 26$	$warp\_lnch\_ovh = 10$	$n_{reg} = 65536$
$n_{fu} = 4$	$l_m = 2$	$block\_lnch\_ovh = 382$	
$warp\_size = 32$	$l_b = 0$	$issue\_cycle\ = 1$	

$$w\ (eq.\ 6) = \left\lceil \frac{256}{32 \times 4} \right\rceil = 2$$

Table 6-1, Latency, FUs, Throughput, u, d and p for KNN on 940MX

Inst#	Instructions	FU	n^i_{fu}	T_i	n^i_{ma}	Latency		u	d	p
						comp	comm			
1	ld.param.u64 %rd1, [Kernel_param_0];	SPs	32	32		6		15	0	0
2	ld.param.u64 %rd2, [Kernel_param_1];	SPs	32	32		6		16	0	0
3	ld.param.u32 %r2, [Kernel_param_2];	SPs	32	32		6		13	0	0
4	ld.param.f32 %f1, [Kernel_param_3];	SPs	32	32		6		22	0	0
5	ld.param.f32 %f2, [Kernel_param_4];	SPs	32	32		6		24	0	0
6	mov.u32 %r3, %ctaid.y;	SPs	32	32		27		9	0	0
7	mov.u32 %r4, %nctaid.x;	SPs	32	32		6		9	0	0
8	mov.u32 %r5, %ctaid.x;	SPs	32	32		27		9	1	0
9	mad.lo.s32 %r6, %r3, %r4, %r5;	SPs	32	32		6		12	0	0
10	mov.u32 %r7, %ntid.x;	SPs	32	32		6		12	0	0
11	mov.u32 %r8, %tid.x;	SPs	32	32		27		12	0	0
12	mad.lo.s32 %r1, %r6, %r7, %r8;	SPs	32	32		6		13	1	0
13	setp.ge.s32 %p1, %r1, %r2;	SPs	32	16		6		14	1	0
14	@%p1 bra BB0_2;	SPs	32	16		6		0	0	0
15	cvta.to.global.u64 %rd3, %rd1;	SPs	32	8		6		20	0	0
16	cvta.to.global.u64 %rd4, %rd2;	SPs	32	8		6		18	0	0
17	mul.wide.s32 %rd5, %r1, 4;	SPs	32	32		6		18	1	0
18	add.s64 %rd6, %rd4, %rd5;	SPs	32	32		6		28	0	0
19	mul.wide.s32 %rd7, %r1, 8;	SPs	32	32		6		20	1	0
20	add.s64 %rd8, %rd3, %rd7;	SPs	32	32		6		21	1	1
21	ld.global.f32 %f3, [%rd8];	LDST	8	8	2		313	22	1	1
22	sub.f32 %f4, %f1, %f3;	SPs	32	32		6		26	0	1
23	ld.global.f32 %f5, [%rd8+4];	LDST	8	8	2	19[1]		24	1	1
24	sub.f32 %f6, %f2, %f5;	SPs	32	32		6		25	1	0
25	mul.f32 %f7, %f6, %f6;	SPs	32	32		6		26	1	0
26	fma.rn.f32 %f8, %f4, %f4, %f7;	SPs	32	32		20		27	1	1
27	sqrt.rn.f32 %f9, %f8;	SFU	8	8		370		28	1	1
28	st.global.f32 [%rd6], %f9;	LDST	8	8	2		313	0	--	--
--	BB0_2:									
--	ret;									

Table 6-2, BLOCK data structure for KNN on 940MX

Inst#	Instructions	$comp_i$						$comm_i$	ovh_i	$sync_i$	u	d	p
		Issue	FU_1	FU_2	FU_3	FU_4	$32/T_i$						
1	ld.param.u64	2	6+1				1			0	15	0	0
2	ld.param.u64	2	2				1			0	16	0	0
3	ld.param.u32	2	2				1			0	13	0	0
4	ld.param.f32	2	2				1			0	22	0	0
5	ld.param.f32	2	2				1			0	24	0	0
6	mov.u32	2	2				1			0	9	0	0
7	mov.u32	2	2				1			0	9	0	0
8	mov.u32	2	2				1			2	9	1	0
9	mad.lo.s32	2	6+1				1			0	12	0	0
10	mov.u32	2	2				1			0	12	0	0
11	mov.u32	2	2				1			2	12	0	0
12	mad.lo.s32	2	6+1				1			7	13	1	0
13	setp.ge.s32	2	6+2				2			8	14	1	0
14	@%p1 bra BB0_2;	2	6+2				2			1	0	0	0
15	cvta.to.global.u64	2	4				4			0	20	0	0
16	cvta.to.global.u64	2	4				4			0	18	0	0

[1] L1 cache hit

17	mul.wide.s32	2	2				1			2	18	1	0
18	add.s64	2	6+1				1			0	28	0	0
19	mul.wide.s32	2	2				1			2	20	1	0
20	add.s64	2	6+1				1			7	21	1	1
21	ld.global.f32	8					4	1252		1	22	1	1
22	sub.f32		6+1				1			0	26	0	1
23	ld.global.f32	8				19+12	4			31	24	1	1
24	sub.f32	2	6+1				1			7	25	1	0
25	mul.f32	2	6+1				1			7	26	1	0
26	fma.rn.f32	2	6+1				1			7	27	1	1
27	sqrt.rn.f32	8			370+4		4			374	28	1	1
28	st.global.f32	8					4	1252		1	0	--	--
--	BB0_2:												
--	ret;												

Table 6-3, Level 1 and Level 2 supersteps for KNN on 940MX (s_1=2, s_2=15)

Level1:	Superstep#: <start ~ end, comp, comm, ovh>, Itteration 1: <1~14,65,0,0>, 1 2: <15~28,488,2504,0>, 2
Level2:	Superstep#: <start ~ end, comp, comm, ovh> 1: <1~8,23,0,0> 5: <14~14,10,0,0> 9: <21~21,8,1252,0> 13: <26~26,9,0,0> 2: <9~11,13,0,0> 6: <15~17,12,0,0> 10: <22~23,31,0,0> 14: <27~27,382,0,0> 3: <12~12,9,0,0> 7: <18~19,11,0,0> 11: <24~24,9,0,0> 15: <28~28,8,1252,0> 4: <13~13,10,0,0> 8: <20~20,9,0,0> 12: <25~25,9,0,0>

$parallel\_comp\ (eq.10) = 1 \times 65 + 1 \times 488 = 553$

$block\_bar\_ovh\ (eq.11) = 0$

$block\_comm\ (eq.12) = 1 \times 0 + 1 \times 2504 = 2504$

$block\_comm_{\Delta} = 1252$

$COMP\ (eq.5) = 2 \times 10 + 553 = 573$

$warp\_comp = COMP \div w = 573 \div 2 = 287$

$warp\_comm_{\Delta} = 626$

$$warps\_need\ (eq.15) = 4 \times \left(\left\lceil \frac{626 \times 26}{287 \times (2-1)} \right\rceil + 1\right) = 232$$

$$nonoverlapped\ (eq.17) = min\left\{\frac{2504}{2}, 313 + \frac{1252}{2} \times max\left(0{,}1 - \frac{2 \times 4}{232}\right)\right\} = 918$$

$block\_exec\_cycle\ (eq.4) = 382 + 0 + 918 + 573 = 1873$

$comp\ (eq.19) = 573$

$novlp\ (eq.20) = 918$

$$\rho\ (eq.21) = max\left\{1, min\left\{\left\lfloor \frac{2048}{256} \right\rfloor, \left\lfloor \frac{65536}{256 \times 0} \right\rfloor, \left\lfloor \frac{48\ KB}{0\ B} \right\rfloor\right\}\right\} = 8$$

$$\mathrm{K}\ (eq.22) = \frac{168}{4 \times 8} = 5.25$$

$$\tau\ (eq.23) = \left\lceil \frac{918}{573} \right\rceil + 1 = 3$$

$\mu = 1.93$

$$\rho \geq \tau : kernel\_exec\_cycle\ (eq.24) = 382 + \frac{168}{4} \times \frac{573}{1.93} + \frac{918}{2} = 13311$$

$mesured\_kernel\_exec\_cycle = 13887$

$$error\ \% = \frac{|13887 - 13311|}{13887} \times 100 = 4.14\ \%$$

7. Details of results for MM on 940MX

$d = 3$	$n_b = 200$	$g_0 = 19$	$\max\_thread\_per\_sm = 2048$
$n_c = 128$	$n_t = 1024$	$g_1 = 160$	$reg\_per\_thread = 22$
$n_{SM} = 4$	$h = 155$	$g_2 = 134$	$shmem\_size = 48\ KB$
$n_{ws} = 4$	$l = 1163$	$mem\_lat = G_2 = 313$	$shmem\_per\_block = 2048\ B$
$n_{du} = 8$	$l_c = 1122$	$warp\_lnch\_ovh = 10$	$n_{reg} = 65536$
$n_{fu} = 4$	$l_m = 21$	$block\_lnch\_ovh = 382$	
$warp\_size = 32$	$l_b = 20$	$issue\_cycle\ = 1$	

$$w\ (eq.\ 6) = \left\lceil \frac{1024}{32 \times 4} \right\rceil = 8$$

Table 7-1, Latency, FUs, Throughput, u, d and p for MM on 940MX

Inst#	Instructions	FU	n^i_{fu}	T_i	n^i_{ma}	Latency (Cycle)		u	d	p
						محاسبات	ارتباطات			
1	ld.param.u64 %rd7, [Kernel_param_0];	SPs	32	32		6		152	0	0
2	ld.param.u64 %rd8, [Kernel_param_1];	SPs	32	32		6		15	0	0
3	ld.param.u64 %rd9, [Kernel_param_2];	SPs	32	32		6		14	0	0
4	ld.param.u32 %r12, [Kernel_param_3];	SPs	32	32		6		7	0	0
5	ld.param.u32 %r13, [Kernel_param_4];	SPs	32	32		6		10	0	0
6	mov.u32 %r14, %ctaid.y;	SPs	32	32		27		7	1	0
7	mul.lo.s32 %r15, %r12, %r14;	SPs	32	32		6		8	1	0
8	shl.b32 %r30, %r15, 5;	SPs	32	16		6		17	0	0
9	add.s32 %r2, %r12, -1;	SPs	32	32		6		11	0	0
10	shl.b32 %r3, %r13, 5;	SPs	32	16		6		141	0	0
11	setp.lt.s32 %p1, %r2, 0;	SPs	32	16		6		13	0	0
12	mov.f32 %f103, 0f00000000;	SPs	32	32		6		32	0	0
13	@%p1 bra BB0_3;	SPs	32	16		6		0	0	0
14	cvta.to.global.u64 %rd1, %rd9;	SPs	32	8		6		40	0	0
15	cvta.to.global.u64 %rd2, %rd8;	SPs	32	8		6		35	0	0
16	mov.u32 %r16, %tid.x;	SPs	32	32		27		21	0	0
17	add.s32 %r4, %r2, %r30;	SPs	32	32		6		143	0	0
18	mov.u32 %r17, %ctaid.x;	SPs	32	32		27		19	1	0
19	shl.b32 %r29, %r17, 5;	SPs	32	16		6		38	0	0
20	mov.u32 %r18, %tid.y;	SPs	32	32		27		21	1	0
21	mad.lo.s32 %r6, %r18, %r12, %r16;	SPs	32	32		6		33	0	0
22	mul.wide.s32 %rd10, %r18, 128;	SPs	32	32		6		24	0	0
23	mov.u64 %rd11, Kernel $__cuda_...;	SPs	32	32		6		24	1	0
24	add.s64 %rd5, %rd11, %rd10;	SPs	32	32		6		26	0	0
25	mul.wide.s32 %rd12, %r16, 4;	SPs	32	32		6		26	1	0
26	add.s64 %rd3, %rd5, %rd12;	SPs	32	32		6		37	0	0
27	mad.lo.s32 %r7, %r18, %r13, %r16;	SPs	32	32		6		38	0	0
28	mov.u64 %rd13, Kernel $__cuda_...;	SPs	32	32		6		29	1	0
29	add.s64 %rd14, %rd13, %rd10;	SPs	32	32		6		30	1	0
30	add.s64 %rd4, %rd14, %rd12;	SPs	32	32		6		42	0	0
31	add.s64 %rd6, %rd13, %rd12;	SPs	32	32		6		44	0	0
32	mov.f32 %f103, 0f00000000;	SPs	32	32		6		46	0	0
--	BB1_2:							--	--	--
33	add.s32 %r19, %r6, %r30;	SPs	32	32		6		34	1	0
34	mul.wide.s32 %rd15, %r19, 4;	SPs	32	32		6		35	1	0
35	add.s64 %rd16, %rd2, %rd15;	SPs	32	32		6		36	1	1
36	ld.global.f32 %f6, [%rd16];	LDST	8	8	1		313	37	1	0
37	st.shared.f32 [%rd3], %f6;	LDST	8	8	1	20		0	0	1
38	add.s32 %r20, %r7, %r29;	SPs	32	32		6		39	1	0
39	mul.wide.s32 %rd17, %r20, 4;	SPs	32	32		6		40	1	0
40	add.s64 %rd18, %rd1, %rd17;	SPs	32	32		6		41	1	1
41	ld.global.f32 %f7, [%rd18];	LDST	8	8	1		313	42	1	0
42	st.shared.f32 [%rd4], %f7;	LDST	8	8	1	20		0	0	0
43	bar.sync 0;		MI			230		0	0	0
44	ld.shared.f32 %f8, [%rd6];	LDST	8	8	1	6		46	0	0
45	ld.shared.f32 %f9, [%rd5];	LDST	8	8	1	6		46	1	1
46	fma.rn.f32 %f10, %f9, %f8, %f103;	SPs	32	32		20		49	0	1
47	ld.shared.f32 %f11, [%rd6+128];	LDST	8	8	1	6		49	0	0
48	ld.shared.f32 %f12, [%rd5+4];	LDST	8	8	1	6		49	1	1
49	fma.rn.f32 %f13, %f12, %f11, %f10;	SPs	32	32		20		52	0	1
50	ld.shared.f32 %f14, [%rd6+256];	LDST	8	8	1	6		52	0	0
51	ld.shared.f32 %f15, [%rd5+8];	LDST	8	8	1	6		52	1	1
52	fma.rn.f32 %f16, %f15, %f14, %f13;	SPs	32	32		20		55	0	1
53	ld.shared.f32 %f17, [%rd6+384];	LDST	8	8	1	6		55	0	0

54	ld.shared.f32	%f18, [%rd5+12];	LDST	8	8	1	6		55	1	1
55	fma.rn.f32	%f19, %f18, %f17, %f16;	SPs	32	32		20		58	0	1
56	ld.shared.f32	%f20, [%rd6+512];	LDST	8	8	1	6		58	0	0
57	ld.shared.f32	%f21, [%rd5+16];	LDST	8	8	1	6		58	1	1
58	fma.rn.f32	%f22, %f21, %f20, %f19;	SPs	32	32		20		61	0	1
59	ld.shared.f32	%f23, [%rd6+640];	LDST	8	8	1	6		61	0	0
60	ld.shared.f32	%f24, [%rd5+20];	LDST	8	8	1	6		61	1	1
61	fma.rn.f32	%f25, %f24, %f23, %f22;	SPs	32	32		20		64	0	1
62	ld.shared.f32	%f26, [%rd6+768];	LDST	8	8	1	6		64	0	0
63	ld.shared.f32	%f27, [%rd5+24];	LDST	8	8	1	6		64	1	1
64	fma.rn.f32	%f28, %f27, %f26, %f25;	SPs	32	32		20		67	0	1
65	ld.shared.f32	%f29, [%rd6+896];	LDST	8	8	1	6		67	0	0
66	ld.shared.f32	%f30, [%rd5+28];	LDST	8	8	1	6		67	1	1
67	fma.rn.f32	%f31, %f30, %f29, %f28;	SPs	32	32		20		70	0	1
68	ld.shared.f32	%f32, [%rd6+1024];	LDST	8	8	1	6		70	0	0
69	ld.shared.f32	%f33, [%rd5+32];	LDST	8	8	1	6		70	1	1
70	fma.rn.f32	%f34, %f33, %f32, %f31;	SPs	32	32		20		73	0	1
71	ld.shared.f32	%f35, [%rd6+1152];	LDST	8	8	1	6		73	0	0
72	ld.shared.f32	%f36, [%rd5+36];	LDST	8	8	1	6		73	1	1
73	fma.rn.f32	%f37, %f36, %f35, %f34;	SPs	32	32		20		76	0	1
74	ld.shared.f32	%f38, [%rd6+1280];	LDST	8	8	1	6		76	0	0
75	ld.shared.f32	%f39, [%rd5+40];	LDST	8	8	1	6		76	1	1
76	fma.rn.f32	%f40, %f39, %f38, %f37;	SPs	32	32		20		79	0	1
77	ld.shared.f32	%f41, [%rd6+1408];	LDST	8	8	1	6		79	0	0
78	ld.shared.f32	%f42, [%rd5+44];	LDST	8	8	1	6		79	1	1
79	fma.rn.f32	%f43, %f42, %f41, %f40;	SPs	32	32		20		82	0	1
80	ld.shared.f32	%f44, [%rd6+1536];	LDST	8	8	1	6		82	0	0
81	ld.shared.f32	%f45, [%rd5+48];	LDST	8	8	1	6		82	1	1
82	fma.rn.f32	%f46, %f45, %f44, %f43;	SPs	32	32		20		85	0	1
83	ld.shared.f32	%f47, [%rd6+1664];	LDST	8	8	1	6		85	0	0
84	ld.shared.f32	%f48, [%rd5+52];	LDST	8	8	1	6		85	1	1
85	fma.rn.f32	%f49, %f48, %f47, %f46;	SPs	32	32		20		88	0	1
86	ld.shared.f32	%f50, [%rd6+1792];	LDST	8	8	1	6		88	0	0
87	ld.shared.f32	%f51, [%rd5+56];	LDST	8	8	1	6		88	1	1
88	fma.rn.f32	%f52, %f51, %f50, %f49;	SPs	32	32		20		91	0	1
89	ld.shared.f32	%f53, [%rd6+1920];	LDST	8	8	1	6		91	0	0
90	ld.shared.f32	%f54, [%rd5+60];	LDST	8	8	1	6		91	1	1
91	fma.rn.f32	%f55, %f54, %f53, %f52;	SPs	32	32		20		94	0	1
92	ld.shared.f32	%f56, [%rd6+2048];	LDST	8	8	1	6		94	0	0
93	ld.shared.f32	%f57, [%rd5+64];	LDST	8	8	1	6		94	1	1
94	fma.rn.f32	%f58, %f57, %f56, %f55;	SPs	32	32		20		97	0	1
95	ld.shared.f32	%f59, [%rd6+2176];	LDST	8	8	1	6		97	0	0
96	ld.shared.f32	%f60, [%rd5+68];	LDST	8	8	1	6		97	1	1
97	fma.rn.f32	%f61, %f60, %f59, %f58;	SPs	32	32		20		100	0	1
98	ld.shared.f32	%f62, [%rd6+2304];	LDST	8	8	1	6		100	0	0
99	ld.shared.f32	%f63, [%rd5+72];	LDST	8	8	1	6		100	1	1
100	fma.rn.f32	%f64, %f63, %f62, %f61;	SPs	32	32		20		103	0	1
101	ld.shared.f32	%f65, [%rd6+2432];	LDST	8	8	1	6		103	0	0
102	ld.shared.f32	%f66, [%rd5+76];	LDST	8	8	1	6		103	1	1
103	fma.rn.f32	%f67, %f66, %f65, %f64;	SPs	32	32		20		106	0	1
104	ld.shared.f32	%f68, [%rd6+2560];	LDST	8	8	1	6		106	0	0
105	ld.shared.f32	%f69, [%rd5+80];	LDST	8	8	1	6		106	1	1
106	fma.rn.f32	%f70, %f69, %f68, %f67;	SPs	32	32		20		109	0	1
107	ld.shared.f32	%f71, [%rd6+2688];	LDST	8	8	1	6		109	0	0
108	ld.shared.f32	%f72, [%rd5+84];	LDST	8	8	1	6		109	1	1
109	fma.rn.f32	%f73, %f72, %f71, %f70;	SPs	32	32		20		112	0	1
110	ld.shared.f32	%f74, [%rd6+2816];	LDST	8	8	1	6		112	0	0
111	ld.shared.f32	%f75, [%rd5+88];	LDST	8	8	1	6		112	1	1
112	fma.rn.f32	%f76, %f75, %f74, %f73;	SPs	32	32		20		115	0	1
113	ld.shared.f32	%f77, [%rd6+2944];	LDST	8	8	1	6		115	0	0
114	ld.shared.f32	%f78, [%rd5+92];	LDST	8	8	1	6		115	1	1
115	fma.rn.f32	%f79, %f78, %f77, %f76;	SPs	32	32		20		118	0	1
116	ld.shared.f32	%f80, [%rd6+3072];	LDST	8	8	1	6		118	0	0
117	ld.shared.f32	%f81, [%rd5+96];	LDST	8	8	1	6		118	1	1
118	fma.rn.f32	%f82, %f81, %f80, %f79;	SPs	32	32		20		121	0	1
119	ld.shared.f32	%f83, [%rd6+3200];	LDST	8	8	1	6		121	0	0
120	ld.shared.f32	%f84, [%rd5+100];	LDST	8	8	1	6		121	1	1
121	fma.rn.f32	%f85, %f84, %f83, %f82;	SPs	32	32		20		124	0	1
122	ld.shared.f32	%f86, [%rd6+3328];	LDST	8	8	1	6		124	0	0
123	ld.shared.f32	%f87, [%rd5+104];	LDST	8	8	1	6		124	1	1
124	fma.rn.f32	%f88, %f87, %f86, %f85;	SPs	32	32		20		127	0	1
125	ld.shared.f32	%f89, [%rd6+3456];	LDST	8	8	1	6		127	0	0
126	ld.shared.f32	%f90, [%rd5+108];	LDST	8	8	1	6		127	1	1
127	fma.rn.f32	%f91, %f90, %f89, %f88;	SPs	32	32		20		130	0	1

128	ld.shared.f32 %f92, [%rd6+3584];	LDST	8	8	1	6		130	0	0
129	ld.shared.f32 %f93, [%rd5+112];	LDST	8	8	1	6		130	1	1
130	fma.rn.f32 %f94, %f93, %f92, %f91;	SPs	32	32		20		133	0	1
131	ld.shared.f32 %f95, [%rd6+3712];	LDST	8	8	1	6		133	0	0
132	ld.shared.f32 %f96, [%rd5+116];	LDST	8	8	1	6		133	1	1
133	fma.rn.f32 %f97, %f96, %f95, %f94;	SPs	32	32		20		136	0	1
134	ld.shared.f32 %f98, [%rd6+3840];	LDST	8	8	1	6		136	0	0
135	ld.shared.f32 %f99, [%rd5+120];	LDST	8	8	1	6		136	1	1
136	fma.rn.f32 %f100, %f99, %f98, %f97;	SPs	32	32		20		139	0	1
137	ld.shared.f32 %f101, [%rd6+3968];	LDST	8	8	1	6		139	0	0
138	ld.shared.f32 %f102, [%rd5+124];	LDST	8	8	1	6		139	1	1
139	fma.rn.f32 %f103, %f102, %f101, %f100;	SPs	32	32		20		155	0	0
140	bar.sync 0;		MI			230		0	0	0
141	add.s32 %r29, %r29, %r3;	SPs	32	32		6		0	0	0
142	add.s32 %r30, %r30, 32;	SPs	32	32		6		143	1	0
143	setp.le.s32 %p2, %r30, %r4;	SPs	32	16		6		144	1	0
144	@%p2 bra BB1_2;	SPs	32	16		6		0	0	0
--	BB1_3:									
145	mov.u32 %r22, %ctaid.x;	SPs	32	32		27		146	1	0
146	shl.b32 %r23, %r22, 5;	SPs	32	16		6		147	1	0
147	mad.lo.s32 %r24, %r14, %r3, %r23;	SPs	32	32		6		149	0	0
148	mov.u32 %r25, %tid.x;	SPs	32	32		27		149	1	0
149	add.s32 %r26, %r24, %r25;	SPs	32	32		6		151	0	0
150	mov.u32 %r27, %tid.y;	SPs	32	32		27		151	1	0
151	mad.lo.s32 %r28, %r27, %r13, %r26;	SPs	32	32		6		153	0	1
152	cvta.to.global.u64 %rd19, %rd7;	SPs	32	8		6		154	0	1
153	mul.wide.s32 %rd20, %r28, 4;	SPs	32	32		6		154	1	0
154	add.s64 %rd21, %rd19, %rd20;	SPs	32	32		6		155	1	1
155	st.global.f32 [%rd21], %f103;	LDST	8	8	1		313	0	--	--
--	ret;									

Table 7-2, BLOCK data structure for MM on 940MX

Inst#	Instructions	$comp_i$						$comm_i$	ovh_i	$sync_i$	u	d	p
		Issue	FU_1	FU_2	FU_3	FU_4	$32/T_i$						
1	ld.param.u64	8	6+7				1			0	152	0	0
2	ld.param.u64	8	8				1			0	15	0	0
3	ld.param.u64	8	8				1			0	14	0	0
4	ld.param.u32	8	8				1			0	7	0	0
5	ld.param.u32	8	8				1			0	10	0	0
6	mov.u32	8	8				1			8	7	1	0
7	mul.lo.s32	8	6+7				1			13	8	1	0
8	shl.b32	8	6+14				2			0	17	0	0
9	add.s32	8	8				1			12	11	0	0
10	shl.b32	8	16				2			0	141	0	0
11	setp.lt.s32	8	16				2			16	13	0	0
12	mov.f32	8	8				1			0	32	0	0
13	@%p1 bra BB0_3;	8	16				2			0	0	0	0
14	cvta.to.global.u64	8	32				4			0	40	0	0
15	cvta.to.global.u64	8	32				4			0	35	0	0
16	mov.u32	8	8				1			0	21	0	0
17	add.s32	8	8				1			0	143	0	0
18	mov.u32	8	8				1			8	19	1	0
19	shl.b32	8	6+14				2			0	38	0	0
20	mov.u32	8	8				1			8	21	1	0
21	mad.lo.s32	8	6+7				1			0	33	0	0
22	mul.wide.s32	8	8				1			0	24	0	0
23	mov.u64	8	8				1			8	24	1	0
24	add.s64	8	6+7				1			0	26	0	0
25	mul.wide.s32	8	8				1			8	26	1	0
26	add.s64	8	6+7				1			0	37	0	0
27	mad.lo.s32	8	8				1			0	38	0	0
28	mov.u64	8	8				1			8	29	1	0
29	add.s64	8	6+7				1			13	30	1	0
30	add.s64	8	6+7				1			0	42	0	0
31	add.s64	8	8				1			0	44	0	0
32	mov.f32	8	8				1			1	46	0	0
--	BB1_2:										--	--	--
33	add.s32	8	8				1			8	34	1	0
34	mul.wide.s32	8	6+7				1			13	35	1	0
35	add.s64	8	6+7				1			13	36	1	1
36	ld.global.f32	32					4	2504		1	37	1	0
37	st.shared.f32	32				20+28	4			0	0	0	1
38	add.s32		6+7				1			13	39	1	0

39	mul.wide.s32	8	6+7				1			13	40	1	0
40	add.s64	8	6+7				1			13	41	1	1
41	ld.global.f32	32					4	2504		1	42	1	0
42	st.shared.f32	32				20+28	4			0	0	0	0
43	bar.sync 0;	8							230	1	0	0	0
44	ld.shared.f32	32				6+28	4			0	46	0	0
45	ld.shared.f32	32				32	4			32	46	1	1
46	fma.rn.f32		20+7				1			0	49	0	1
47	ld.shared.f32	32				6+28	4			0	49	0	0
48	ld.shared.f32	32				32	4			32	49	1	1
49	fma.rn.f32		20+7				1			0	52	0	1
50	ld.shared.f32	32				6+28	4			0	52	0	0
51	ld.shared.f32	32				32	4			32	52	1	1
52	fma.rn.f32		20+7				1			0	55	0	1
53	ld.shared.f32	32				6+28	4			0	55	0	0
54	ld.shared.f32	32				32	4			32	55	1	1
55	fma.rn.f32		20+7				1			0	58	0	1
56	ld.shared.f32	32				6+28	4			0	58	0	0
57	ld.shared.f32	32				32	4			32	58	1	1
58	fma.rn.f32		20+7				1			0	61	0	1
59	ld.shared.f32	32				6+28	4			0	61	0	0
60	ld.shared.f32	32				32	4			32	61	1	1
61	fma.rn.f32		20+7				1			0	64	0	1
62	ld.shared.f32	32				6+28	4			0	64	0	0
63	ld.shared.f32	32				32	4			32	64	1	1
64	fma.rn.f32		20+7				1			0	67	0	1
65	ld.shared.f32	32				6+28	4			0	67	0	0
66	ld.shared.f32	32				32	4			32	67	1	1
67	fma.rn.f32		20+7				1			0	70	0	1
68	ld.shared.f32	32				6+28	4			0	70	0	0
69	ld.shared.f32	32				32	4			32	70	1	1
70	fma.rn.f32		20+7				1			0	73	0	1
71	ld.shared.f32	32				6+28	4			0	73	0	0
72	ld.shared.f32	32				32	4			32	73	1	1
73	fma.rn.f32		20+7				1			0	76	0	1
74	ld.shared.f32	32				6+28	4			0	76	0	0
75	ld.shared.f32	32				32	4			32	76	1	1
76	fma.rn.f32		20+7				1			0	79	0	1
77	ld.shared.f32	32				6+28	4			0	79	0	0
78	ld.shared.f32	32				32	4			32	79	1	1
79	fma.rn.f32		20+7				1			0	82	0	1
80	ld.shared.f32	32				6+28	4			0	82	0	0
81	ld.shared.f32	32				32	4			32	82	1	1
82	fma.rn.f32		20+7				1			0	85	0	1
83	ld.shared.f32	32				6+28	4			0	85	0	0
84	ld.shared.f32	32				32	4			32	85	1	1
85	fma.rn.f32		20+7				1			0	88	0	1
86	ld.shared.f32	32				6+28	4			0	88	0	0
87	ld.shared.f32	32				32	4			32	88	1	1
88	fma.rn.f32		20+7				1			0	91	0	1
89	ld.shared.f32	32				6+28	4			0	91	0	0
90	ld.shared.f32	32				32	4			32	91	1	1
91	fma.rn.f32		20+7				1			0	94	0	1
92	ld.shared.f32	32				6+28	4			0	94	0	0
93	ld.shared.f32	32				32	4			32	94	1	1
94	fma.rn.f32		20+7				1			0	97	0	1
95	ld.shared.f32	32				6+28	4			0	97	0	0
96	ld.shared.f32	32				32	4			32	97	1	1
97	fma.rn.f32		20+7				1			0	100	0	1
98	ld.shared.f32	32				6+28	4			0	100	0	0
99	ld.shared.f32	32				32	4			32	100	1	1
100	fma.rn.f32		20+7				1			0	103	0	1
101	ld.shared.f32	32				6+28	4			0	103	0	0
102	ld.shared.f32	32				32	4			32	103	1	1
103	fma.rn.f32		20+7				1			0	106	0	1
104	ld.shared.f32	32				6+28	4			0	106	0	0
105	ld.shared.f32	32				32	4			32	106	1	1
106	fma.rn.f32		20+7				1			0	109	0	1
107	ld.shared.f32	32				6+28	4			0	109	0	0
108	ld.shared.f32	32				32	4			32	109	1	1
109	fma.rn.f32		20+7				1			0	112	0	1
110	ld.shared.f32	32				6+28	4			0	112	0	0
111	ld.shared.f32	32				32	4			32	112	1	1
112	fma.rn.f32		20+7				1			0	115	0	1

113	ld.shared.f32	32				6+28	4			0	115	0	0
114	ld.shared.f32	32				32	4			32	115	1	1
115	fma.rn.f32		20+7				1			0	118	0	1
116	ld.shared.f32	32				6+28	4			0	118	0	0
117	ld.shared.f32	32				32	4			32	118	1	1
118	fma.rn.f32		20+7				1			0	121	0	1
119	ld.shared.f32	32				6+28	4			0	121	0	0
120	ld.shared.f32	32				32	4			32	121	1	1
121	fma.rn.f32		20+7				1			0	124	0	1
122	ld.shared.f32	32				6+28	4			0	124	0	0
123	ld.shared.f32	32				32	4			32	124	1	1
124	fma.rn.f32		20+7				1			0	127	0	1
125	ld.shared.f32	32				6+28	4			0	127	0	0
126	ld.shared.f32	32				32	4			32	127	1	1
127	fma.rn.f32		20+7				1			0	130	0	1
128	ld.shared.f32	32				6+28	4			0	130	0	0
129	ld.shared.f32	32				32	4			32	130	1	1
130	fma.rn.f32		20+7				1			0	133	0	1
131	ld.shared.f32	32				6+28	4			0	133	0	0
132	ld.shared.f32	32				32	4			32	133	1	1
133	fma.rn.f32		20+7				1			0	136	0	1
134	ld.shared.f32	32				6+28	4			0	136	0	0
135	ld.shared.f32	32				32	4			32	136	1	1
136	fma.rn.f32		20+7				1			0	139	0	1
137	ld.shared.f32	32				6+28	4			0	139	0	0
138	ld.shared.f32	32				32	4			32	139	1	1
139	fma.rn.f32	8	20+7				1			0	155	0	0
140	bar.sync 0;								230	1	0	0	0
141	add.s32	8	6+7				1			0	0	0	0
142	add.s32	8	8				1			8	143	1	0
143	setp.le.s32	8	6+7				2			13	144	1	0
144	@%p2 bra BB1_2;	8	6+7				2			1	0	0	0
--	BB1_3:												
145	mov.u32	8	8				1			8	146	1	0
146	shl.b32	8	6+14				2			20	147	1	0
147	mad.lo.s32	8	6+7				1			0	149	0	0
148	mov.u32	8	8				1			8	149	1	0
149	add.s32	8	6+7				1			0	151	0	0
150	mov.u32	8	8				1			8	151	1	0
151	mad.lo.s32		6+7				1			5	153	0	1
152	cvta.to.global.u64	8	8				4			0	154	0	1
153	mul.wide.s32	8	8				1			8	154	1	0
154	add.s64	8	6+7				1			13	155	1	1
155	st.global.f32	32					4	2504		1	0	--	--
--	ret;												

Table 7-3, Level 1 and Level 2 supersteps for MM on 940MX (s_1=3, s_2=64)

Level1: Superstep#: <start ~ end, comp, comm, ovh>, Itteration

1: <1~32,475,0,0>, 1 2: <33~144,2574,5008,460>, 10 3: <145~155,192,2504,0>, 1

Level2: Superstep#: <start ~ end, comp, comm, ovh>

```
1: <1~6,61,0,0>          17: <39~39,21,0,0>       33: <79~81,66,0,0>       49: <127~129,66,0,0>
2: <7~7,21,0,0>          18: <40~40,21,0,0>       34: <82~84,66,0,0>       50: <130~132,66,0,0>
3: <8~9,36,0,0>          19: <41~41,32,2504,0>    35: <85~87,66,0,0>       51: <133~135,66,0,0>
4: <10~11,40,0,0>        20: <42~43,80,0,230>     36: <88~90,66,0,0>       52: <136~138,66,0,0>
5: <12~18,120,0,0>       21: <44~45,98,0,0>       37: <91~93,66,0,0>       53: <139~140,35,0,230>
6: <19~20,36,0,0>        22: <46~48,66,0,0>       38: <94~96,66,0,0>       54: <141~142,29,0,0>
7: <21~23,37,0,0>        23: <49~51,66,0,0>       39: <97~99,66,0,0>       55: <143~143,21,0,0>
8: <24~25,29,0,0>        24: <52~54,66,0,0>       40: <100~102,66,0,0>     56: <144~144,21,0,0>
9: <26~28,37,0,0>        25: <55~57,66,0,0>       41: <103~105,66,0,0>     57: <145~145,16,0,0>
10: <29~29,21,0,0>       26: <58~60,66,0,0>       42: <106~108,66,0,0>     58: <146~146,28,0,0>
11: <30~32,37,0,0>       27: <61~63,66,0,0>       43: <109~111,66,0,0>     59: <147~148,29,0,0>
12: <33~33,16,0,0>       28: <64~66,66,0,0>       44: <112~114,66,0,0>     60: <149~150,29,0,0>
13: <34~34,21,0,0>       29: <67~69,66,0,0>       45: <115~117,66,0,0>     61: <151~152,21,0,0>
14: <35~35,21,0,0>       30: <70~72,66,0,0>       46: <118~120,66,0,0>     62: <153~153,16,0,0>
15: <36~36,32,2504,0>    31: <73~75,66,0,0>       47: <121~123,66,0,0>     63: <154~154,21,0,0>
16: <37~38,80,0,0>       32: <76~78,66,0,0>       48: <124~126,66,0,0>     64: <155~155,32,2504,0>
```

$parallel\_comp\ (eq.10) = 1 \times 475 + 10 \times 2574 + 1 \times 192 = 26407$

$block\_bar\_ovh\ (eq.11) = 1 \times 0 + 10 \times 460 + 1 \times 0 = 4600$

$block\_comm\ (eq.12) = 1 \times 0 + 10 \times 5008 + 1 \times 2504 = 52584$

$block\_comm_{\Delta} = 50080$

$COMP\ (eq.5) = 8 \times 10 + 26407 = 26487$

$warp\_comp = 26487 \div 8 = 3311$

$warp\_comm_{\Delta} = 6260$

$$warps\_need\ (eq.15) = 4 \times \left(\left\lceil \frac{6260 \times 1122}{3311 \times (21-1)} \right\rceil + 1\right) = 432$$

$$nonoverlapped\ (eq.17) = \min\left\{\frac{52584}{8}, 313 + \frac{50080}{8} \times \max\left(0{,}1 - \frac{8 \times 4}{432}\right)\right\} = 6110$$

$block\_exec\_cycle\ (eq.4) = 382 + 4600 + 26627 + 6110 = 37719$

$comp\ (eq.19) = 31087$

$novlp\ (eq.20) = 6110$

$$\rho\ (eq.21) = max\left\{1, min\left\{\left\lfloor \frac{2048}{1024} \right\rfloor, \left\lfloor \frac{65536}{1024 \times 22} \right\rfloor, \left\lfloor \frac{48\ KB}{2048\ B} \right\rfloor\right\}\right\} = 2$$

$$\mathrm{K}\ (eq.22) = \frac{200}{4 \times 2} = 25$$

$$\tau\ (eq.23) = \left\lceil \frac{6110}{31087} \right\rceil + 1 = 2$$

$\mu = 1.93$

$$\rho \geq \tau : kernel\_exec\_cycle\ (eq.24) = 382 + \frac{200}{4} \times \frac{31087}{1.5} + \frac{6110}{2} = 1039671$$

$mesured\_kernel\_exec\_cycle = 1185952$

$$error\ \% = \frac{|1185952 - 1039671|}{1185952} \times 100 = 12.33\ \%$$

8. Details of results for Hotspot on GTX 1070

$d = 3$ $n_b = 1849$ $g_0 = 19$ $\max\_thread\_per\_sm = 2048$

$n_c = 128$ $n_t = 256$ $g_1 = 207$ $reg\_per\_thread = 34$

$n_{SM} = 15$ $h = 196$ $g_2 = 168$ $shmem\_size = 48\ KB$

$n_{ws} = 4$ $l = 272$ $mem\_lat = G_2 = 394$ $shmem\_per\_block = 3072\ B$

$n_{du} = 8$ $l_c = 265$ $warp\_lnch\_ovh = 10$ $n_{reg} = 65536$

$n_{fu} = 4$ $l_m = 3$ $block\_lnch\_ovh = 358$

$warp\_size = 32$ $l_b = 4$ $issue\_cycle\ = 1$

$$w\ (eq.\ 6) = \left\lceil \frac{256}{32 \times 4} \right\rceil = 2$$

Table 8-1, Latency, FUs, Throughput, u, d and p for Hotspot on GTX 1070

Inst#	Instructions	FU	n^i_{fu}	T_i	n^i_{ma}	Latency		u	d	p
						Comp.	Comm.			
1	ld.param.u32 %r9, [_Kerrnel_param_0];	SPs	32	32		6		15	0	0
2	ld.param.u64 %rd3, [_Kerrnel_param_1];	SPs	32	32		6		49	0	0
3	ld.param.u64 %rd4, [_Kerrnel_param_2];	SPs	32	32		6		39	0	0
4	ld.param.u64 %rd5, [_Kerrnel_param_3];	SPs	32	32		6		193	0	0
5	ld.param.u32 %r10, [_Kerrnel_param_4];	SPs	32	32		6		32	0	0
6	ld.param.u32 %r11, [_Kerrnel_param_5];	SPs	32	32		6		28	0	0
7	ld.param.u32 %r12, [_Kerrnel_param_6];	SPs	32	32		6		22	0	0
8	ld.param.u32 %r13, [_Kerrnel_param_7];	SPs	32	32		6		20	0	0
9	ld.param.f32 %f2, [_Kerrnel_param_8];	SPs	32	32		6		59	0	0
10	ld.param.f32 %f3, [_Kerrnel_param_9];	SPs	32	32		6		60	0	0
11	ld.param.f32 %f4, [_Kerrnel_param_10];	SPs	32	32		6		61	0	0
12	ld.param.f32 %f5, [_Kerrnel_param_11];	SPs	32	32		6		62	0	0
13	ld.param.f32 %f6, [_Kerrnel_param_12];	SPs	32	32		6		59	0	0
14	mov.u32 %r14, %ctaid.x;	SPs	32	32		29		21	0	0
15	shl.b32 %r15, %r9, 1;	SPs	32	16		6		17	0	0
16	mov.u32 %r16, 16;	SPs	32	32		6		17	1	0
17	sub.s32 %r17, %r16, %r15;	SPs	32	32		6		19	0	0
18	mov.u32 %r18, %ctaid.y;	SPs	32	32		29		19	1	0
19	mul.lo.s32 %r19, %r18, %r17;	SPs	32	32		6		20	1	0
20	sub.s32 %r20, %r19, %r13;	SPs	32	32		6		24	0	0
21	mul.lo.s32 %r21, %r14, %r17;	SPs	32	32		6		22	1	0
22	sub.s32 %r22, %r21, %r12;	SPs	32	32		6		26	0	0
23	mov.u32 %r23, %tid.y;	SPs	32	32		29		24	1	0
24	add.s32 %r1, %r20, %r23;	SPs	32	32		6		27	0	0
25	mov.u32 %r24, %tid.x;	SPs	32	32		29		26	1	0
26	add.s32 %r2, %r22, %r24;	SPs	32	32		6		33	0	0
27	setp.gt.s32 %p1, %r1, -1;	SPs	32	16		6		30	0	0
28	add.s32 %r25, %r11, -1;	SPs	32	32		6		29	1	0
29	setp.le.s32 %p2, %r1, %r25;	SPs	32	16		6		30	1	0
30	and.pred %p3, %p1, %p2;	SPs	32	32		6		35	0	0
31	setp.gt.s32 %p4, %r2, -1;	SPs	32	16		6		34	0	0
32	add.s32 %r26, %r10, -1;	SPs	32	32		6		33	1	0
33	setp.le.s32 %p5, %r2, %r26;	SPs	32	16		6		34	1	0
34	and.pred %p6, %p4, %p5;	SPs	32	32		6		35	1	0
35	and.pred %p7, %p3, %p6;	SPs	32	32		6		36	1	0
36	@!%p7 bra BB0_2;	SPs	32	16		6		0	0	0
37	bra.uni BB0_1;	SPs	32	32		6		0	0	0
--	BB0_1:							--	--	--
38	mad.lo.s32 %r27, %r1, %r10, %r2;	SPs	32	32		6		40	0	0
39	cvta.to.global.u64 %rd6, %rd4;	SPs	32	8		6		41	0	0
40	mul.wide.s32 %rd7, %r27, 4;	SPs	32	32		6		41	1	0
41	add.s64 %rd8, %rd6, %rd7;	SPs	32	32		6		42	1	1
42	ld.global.f32 %f7, [%rd8];	LDST	8	8	1~2		394	48	0	1
43	mul.wide.s32 %rd9, %r23, 64;	SPs	32	32		6		45	0	0
44	mov.u64 %rd10, Kerrnel$__cuda_...;	SPs	32	32		6		45	1	0
45	add.s64 %rd11, %rd10, %rd9;	SPs	32	32		6		47	0	0
46	mul.wide.s32 %rd12, %r24, 4;	SPs	32	32		6		47	1	0
47	add.s64 %rd13, %rd11, %rd12;	SPs	32	32		6		48	1	1
48	st.shared.f32 [%rd13], %f7;	LDST	8	8	1	20		0	0	1
49	cvta.to.global.u64 %rd14, %rd3;	SPs	32	8		6		50	1	0
50	add.s64 %rd15, %rd14, %rd7;	SPs	32	32		6		51	1	1
51	ld.global.f32 %f8, [%rd15];	LDST	8	8	1~2		394	55	0	1
52	mov.u64 %rd16, _Kerrnel$__cuda_...;	SPs	32	32		6		53	1	0
53	add.s64 %rd17, %rd16, %rd9;	SPs	32	32		6		54	1	0
54	add.s64 %rd18, %rd17, %rd12;	SPs	32	32		6		55	1	1

55	st.shared.f32 [%rd18], %f8;	LDST	8	8	1	20		0	0	0
--	BB0_2:							--	--	--
56	bar.sync 0;		MI	--		118		0	0	0
57	setp.lt.s32 %p8, %r9, 1;	SPs	32	16		6		58	1	0
58	@%p8 bra BB0_11;	SPs	32	16		6		0	0	0
59	div.rn.f32 %f9, %f6, %f2;	SFU	8	8		133		63	0	0
60	rcp.rn.f32 %f10, %f3;	SFU	8	8		366		93	0	0
61	rcp.rn.f32 %f11, %f4;	SFU	8	8		366		79	0	0
62	rcp.rn.f32 %f1, %f5;	SFU	8	8		366		158	0	1
63	cvt.f64.f32 %fd1, %f9;	SPs	32	1		6		161	0	0
64	add.s32 %r37, %r20, 15;	SPs	32	32		6		65	1	0
65	setp.gt.s32 %p9, %r37, %r25;	SPs	32	16		6		70	0	0
66	mov.u32 %r39, -15;	SPs	32	32		6		67	1	0
67	sub.s32 %r40, %r39, %r20;	SPs	32	32		6		68	1	0
68	add.s32 %r41, %r11, %r40;	SPs	32	32		6		69	1	0
69	add.s32 %r42, %r41, 14;	SPs	32	32		6		70	1	0
70	selp.b32%r43, %r42, 15, %p9;	SPs	32	16		6		71	1	0
71	setp.lt.s32 %p10, %r23, %r43;	SPs	32	16		6		73	0	0
72	add.s32 %r45, %r23, 1;	SPs	32	32		6		73	1	0
73	selp.b32%r46, %r45, %r43, %p10;	SPs	32	16		6		74	1	0
74	mul.wide.s32 %rd19, %r46, 64;	SPs	32	32		6		76	0	0
75	mov.u64 %rd20, _Kerrnel$__cuda_...;	SPs	32	32		6		76	1	0
76	add.s64 %rd21, %rd20, %rd19;	SPs	32	32		6		78	0	0
77	mul.wide.s32 %rd22, %r24, 4;	SPs	32	32		6		78	1	0
78	add.s64 %rd1, %rd21, %rd22;	SPs	32	32		6		139	0	0
79	cvt.f64.f32 %fd2, %f11;	SPs	32	1		6		144	0	0
80	add.s32 %r51, %r22, 15;	SPs	32	32		6		81	1	0
81	setp.gt.s32 %p11, %r51, %r26;	SPs	32	16		6		85	0	0
82	sub.s32 %r53, %r39, %r22;	SPs	32	32		6		83	1	0
83	add.s32 %r54, %r10, %r53;	SPs	32	32		6		84	1	0
84	add.s32 %r55, %r54, 14;	SPs	32	32		6		85	1	0
85	selp.b32%r56, %r55, 15, %p11;	SPs	32	16		6		86	1	0
86	setp.lt.s32 %p12, %r24, %r56;	SPs	32	16		6		88	0	0
87	add.s32 %r57, %r24, 1;	SPs	32	32		6		88	1	0
88	selp.b32%r58, %r57, %r56, %p12;	SPs	32	16		6		89	1	0
89	mul.wide.s32 %rd23, %r58, 4;	SPs	32	32		6		92	0	0
90	mul.wide.s32 %rd24, %r23, 64;	SPs	32	32		6		91	1	0
91	add.s64 %rd25, %rd20, %rd24;	SPs	32	32		6		92	1	0
92	add.s64 %rd2, %rd25, %rd23;	SPs	32	32		6		151	0	0
93	cvt.f64.f32 %fd3, %f10;	SPs	32	1		6		155	0	0
94	mov.u32 %r59, 1;	SPs	32	32		6		95	1	0
95	sub.s32 %r129, %r59, %r9;	SPs	32	32		6		169	0	0
96	mov.u32 %r130, 0;	SPs	32	32		6		98	0	0
--	BB0_4:							--	--	--
97	mov.u32 %r61, 14;	SPs	32	32		6		98	1	0
98	sub.s32 %r6, %r61, %r130;	SPs	32	32		6		99	1	0
99	setp.le.s32 %p13, %r24, %r6;	SPs	32	16		6		102	0	0
100	add.s32 %r130, %r130, 1;	SPs	32	32		6		101	1	0
101	setp.ge.s32 %p14, %r24, %r130;	SPs	32	16		6		102	1	0
102	and.pred %p15, %p13, %p14;	SPs	32	32		6		104	0	0
103	mov.u16 %rs8, 0;	SPs	32	32		6		171	0	0
104	@!%p15 bra BB0_7;	SPs	32	16		6		0	0	0
105	bra.uni BB0_5;	SPs	32	32		6		0	0	0
--	BB0_5:							--	--	--
106	setp.gt.s32 %p16, %r23, %r6;	SPs	32	16		6		108	0	0
107	setp.lt.s32 %p17, %r23, %r130;	SPs	32	16		6		108	1	0
108	or.pred %p18, %p17, %p16;	SPs	32	32		6		115	0	0
109	shr.s32 %r69, %r22, 31;	SPs	32	16		6		112	0	0
110	setp.gt.s32 %p20, %r24, %r56;	SPs	32	16		6		114	0	0
111	neg.s32 %r78, %r22;	SPs	32	16		6		112	1	0
112	and.b32 %r79, %r69, %r78;	SPs	32	32		6		113	1	0
113	setp.lt.s32 %p21, %r24, %r79;	SPs	32	16		6		114	1	0
114	or.pred %p22, %p21, %p20;	SPs	32	32		6		115	1	0
115	or.pred %p23, %p18, %p22;	SPs	32	32		6		122	0	0
116	shr.s32 %r83, %r20, 31;	SPs	32	16		6		119	0	0
117	setp.gt.s32 %p25, %r23, %r43;	SPs	32	16		6		121	0	0
118	neg.s32 %r90, %r20;	SPs	32	16		6		119	1	0
119	and.b32 %r91, %r83, %r90;	SPs	32	32		6		120	1	0
120	setp.lt.s32 %p26, %r23, %r91;	SPs	32	16		6		121	1	0
121	or.pred %p27, %p26, %p25;	SPs	32	32		6		122	1	0
122	or.pred %p28, %p23, %p27;	SPs	32	32		6		123	1	0
123	@%p28 bra BB0_7;	SPs	32	16		6		0	0	0
124	add.s64 %rd30, %rd25, %rd22;	SPs	32	32		6		125	1	1
125	ld.shared.f32 %f12, [%rd30];	LDST	8	8	1	6		126	1	1

126	cvt.f64.f32 %fd4, %f12;	SPs	32	1		6		142	0	0
127	mov.u64 %rd31, Kerrnel$__cuda_...;	SPs	32	32		6		128	1	0
128	add.s64 %rd32, %rd31, %rd24;	SPs	32	32		6		129	1	0
129	add.s64 %rd33, %rd32, %rd22;	SPs	32	32		6		130	1	1
130	ld.shared.f32 %f13, [%rd33];	LDST	8	8	1	6		131	1	1
131	cvt.f64.f32 %fd5, %f13;	SPs	32	1		6		144	0	0
132	setp.gt.s32 %p29, %r23, %r91;	SPs	32	16		6		134	0	0
133	add.s32 %r103, %r23, -1;	SPs	32	32		6		134	1	0
134	selp.b32 %r104, %r103, %r91, %p29;	SPs	32	16		6		135	1	0
135	mul.wide.s32 %rd34, %r104, 64;	SPs	32	32		6		136	1	0
136	add.s64 %rd35, %rd20, %rd34;	SPs	32	32		6		137	1	0
137	add.s64 %rd36, %rd35, %rd22;	SPs	32	32		6		138	1	1
138	ld.shared.f32 %f14, [%rd36];	LDST	8	8	1	6		140	0	0
139	ld.shared.f32 %f15, [%rd1];	LDST	8	8	1	6		140	1	1
140	add.f32 %f16, %f15, %f14;	SPs	32	32		6		141	1	0
141	cvt.f64.f32 %fd6, %f16;	SPs	32	1		6		143	0	1
142	add.f64 %fd7, %fd4, %fd4;	DPU	1	1		6		143	1	0
143	sub.f64 %fd8, %fd6, %fd7;	DPU	1	1		6		144	1	0
144	fma.rn.f64 %fd9, %fd2, %fd8, %fd5;	DPU	1	1		63		155	0	1
145	setp.gt.s32 %p30, %r24, %r79;	SPs	32	16		6		147	0	0
146	add.s32 %r111, %r24, -1;	SPs	32	32		6		147	1	0
147	selp.b32 %r112, %r111, %r79, %p30;	SPs	32	16		6		148	1	0
148	mul.wide.s32 %rd37, %r112, 4;	SPs	32	32		6		149	1	0
149	add.s64 %rd38, %rd25, %rd37;	SPs	32	32		6		150	1	1
150	ld.shared.f32 %f17, [%rd38];	LDST	8	8	1	6		152	0	0
151	ld.shared.f32 %f18, [%rd2];	LDST	8	8	1	6		152	1	1
152	add.f32 %f19, %f18, %f17;	SPs	32	32		6		153	1	0
153	cvt.f64.f32 %fd10, %f19;	SPs	32	1		6		154	1	0
154	sub.f64 %fd11, %fd10, %fd7;	DPU	1	1		6		155	1	0
155	fma.rn.f64 %fd12, %fd3, %fd11, %fd9;	DPU	1	1		63		160	0	1
156	mov.f32 %f20, 0f42A00000;	SPs	32	32		6		157	1	0
157	sub.f32 %f21, %f20, %f12;	SPs	32	32		6		158	1	0
158	mul.f32 %f22, %f1, %f21;	SPs	32	32		6		159	1	0
159	cvt.f64.f32 %fd13, %f22;	SPs	32	1		6		160	1	0
160	add.f64 %fd14, %fd13, %fd12;	DPU	1	1		6		161	1	0
161	fma.rn.f64 %fd15, %fd1, %fd14, %fd4;	DPU	1	1		63		162	1	1
162	cvt.rn.f32.f64 %f23, %fd15;	SPs	32	8		6		166	0	0
163	mov.u64 %rd39, _Kerrnel$__cuda_...;	SPs	32	32		6		164	1	0
164	add.s64 %rd40, %rd39, %rd24;	SPs	32	32		6		165	1	0
165	add.s64 %rd41, %rd40, %rd22;	SPs	32	32		6		166	1	1
166	st.shared.f32 [%rd41], %f23;	LDST	8	8	1	20		0	0	1
167	mov.u16 %rs8, 1;	SPs	32	32		6		171	0	0
--	BB0_7:							--	--	--
168	bar.sync 0;		MI			118		0	0	0
169	setp.eq.s32 %p31, %r129, 0;	SPs	32	16		6		170	1	0
170	@%p31 bra BB0_11;	SPs	32	16		6		0	0	0
171	setp.eq.s16 %p32, %rs8, 0;	SPs	32	16		6		172	1	0
172	@%p32 bra BB0_10;	SPs	32	16		6		0	0	0
173	mov.u64 %rd43, _Kerrnel$__cuda_...;	SPs	32	32		6		174	1	0
174	add.s64 %rd44, %rd43, %rd24;	SPs	32	32		6		175	1	0
175	add.s64 %rd46, %rd44, %rd22;	SPs	32	32		6		176	1	1
176	ld.shared.f32 %f24, [%rd46];	LDST	8	8	1	6		178	0	1
177	add.s64 %rd49, %rd25, %rd22;	SPs	32	32		6		178	1	1
178	st.shared.f32 [%rd49], %f24;	LDST	8	8	1	20		0	0	0
--	BB0_10:							--	--	--
179	bar.sync 0;		MI			118		0	0	0
180	add.s32 %r129, %r129, 1;	SPs	32	32		6		0	0	0
181	setp.lt.s32 %p33, %r130, %r9;	SPs	32	16		6		182	1	0
182	@%p33 bra BB0_4;	SPs	32	16		6		0	0	0
--	BB0_11:							--	--	--
183	and.b16 %rs7, %rs8, 255;	SPs	32	32		6		184	1	0
184	setp.eq.s16 %p34, %rs7, 0;	SPs	32	16		6		185	1	0
185	@%p34 bra BB0_13;	SPs	32	16		6		0	0	0
186	mul.wide.s32 %rd50, %r23, 64;	SPs	32	32		6		188	0	0
187	mov.u64 %rd51, _Kerrnel$__cuda_...;	SPs	32	32		6		188	1	0
188	add.s64 %rd52, %rd51, %rd50;	SPs	32	32		6		190	0	0
189	mul.wide.s32 %rd53, %r24, 4;	SPs	32	32		6		190	1	0
190	add.s64 %rd54, %rd52, %rd53;	SPs	32	32		6		191	1	1
191	ld.shared.f32 %f25, [%rd54];	LDST	8	8	1	6		196	0	1
192	mad.lo.s32 %r128, %r1, %r10, %r2;	SPs	32	32		6		194	0	0
193	cvta.to.global.u64 %rd55, %rd5;	SPs	32	8		6		195	0	0
194	mul.wide.s32 %rd56, %r128, 4;	SPs	32	32		6		195	1	0
195	add.s64 %rd57, %rd55, %rd56;	SPs	32	32		6		196	1	1
196	st.global.f32 [%rd57], %f25;	LDST	8	8	1~2		394	0	0	0

--	BB0_13:									
--	ret;									

Table 8-2, BLOCK data structure for Hotspot on GTX 1070

Inst#	Instructions	$comp_i$						$comm_i$	ovh_i	$sync_i$	u	d	p
		Issue	FU_1	FU_2	FU_3	FU_4	$32^1/T_i$						
1	ld.param.u32	2	6+1				1			0	15	0	0
2	ld.param.u64	2	2				1			0	49	0	0
3	ld.param.u64	2	2				1			0	39	0	0
4	ld.param.u64	2	2				1			0	193	0	0
5	ld.param.u32	2	2				1			0	32	0	0
6	ld.param.u32	2	2				1			0	28	0	0
7	ld.param.u32	2	2				1			0	22	0	0
8	ld.param.u32	2	2				1			0	20	0	0
9	ld.param.f32	2	2				1			0	59	0	0
10	ld.param.f32	2	2				1			0	60	0	0
11	ld.param.f32	2	2				1			0	61	0	0
12	ld.param.f32	2	2				1			0	62	0	0
13	ld.param.f32	2	2				1			0	59	0	0
14	mov.u32	2	2				1			0	21	0	0
15	shl.b32	2	4				2			0	17	0	0
16	mov.u32	2	2				1			2	17	1	0
17	sub.s32	2	6+1				1			0	19	0	0
18	mov.u32	2	2				1			2	19	1	0
19	mul.lo.s32	2	6+1				1			7	20	1	0
20	sub.s32	2	6+1				1			0	24	0	0
21	mul.lo.s32	2	2				1			2	22	1	0
22	sub.s32	2	6+1				1			0	26	0	0
23	mov.u32	2	2				1			2	24	1	0
24	add.s32	2	6+1				1			0	27	0	0
25	mov.u32	2	2				1			2	26	1	0
26	add.s32	2	6+1				1			0	33	0	0
27	setp.gt.s32	2	4				2			0	30	0	0
28	add.s32	2	2				1			2	29	1	0
29	setp.le.s32	2	6+2				2			8	30	1	0
30	and.pred	2	6+1				1			0	35	0	0
31	setp.gt.s32	2	4				2			0	34	0	0
32	add.s32	2	2				1			2	33	1	0
33	setp.le.s32	2	6+2				2			8	34	1	0
34	and.pred	2	6+1				1			7	35	1	0
35	and.pred	2	6+1				1			7	36	1	0
36	@!%p7 bra BB0_2;	2	6+2				2			1	0	0	0
37	bra.uni BB0_1;	2	2				1			0	0	0	0
--	BB0_1:										--	--	--
38	mad.lo.s32	2	6+1				1			5	40	0	0
39	cvta.to.global.u64	2	8				4			0	41	0	0
40	mul.wide.s32	2	2				1			7	41	1	0
41	add.s64	2	6+1				1			7	42	1	1
42	ld.global.f32	8					--	1576		0	48	0	1
43	mul.wide.s32		2				1			1	45	0	0
44	mov.u64	2	2				1			2	45	1	0
45	add.s64	2	6+1				1			0	47	0	0
46	mul.wide.s32	2	2				1			2	47	1	0
47	add.s64	2	6+1				1			7	48	1	1
48	st.shared.f32	8				20+4	4			0	0	0	1
49	cvta.to.global.u64		8				4			8	50	1	0
50	add.s64	2	6+1				1			7	51	1	1
51	ld.global.f32	8					--	1576		0	55	0	1
52	mov.u64		2				1			1	53	1	0
53	add.s64	2	6+1				1			7	54	1	0
54	add.s64	2	6+1				1			7	55	1	1
55	st.shared.f32	8				20+4	4			0	0	0	0
--	BB0_2:										--	--	--
56	bar.sync	2					--		118	1	0	0	0
57	setp.lt.s32	2	6+2				2			8	58	1	0
58	@%p8 bra BB0_11;	2	6+2				2			1	0	0	0
59	div.rn.f32	8			133+4		4			117	63	0	0
60	rcp.rn.f32	8			8		4			0	93	0	0
61	rcp.rn.f32	8			8		4			0	79	0	0
62	rcp.rn.f32	8			8		4			0	158	0	1

[1] warp_size

63	cvt.f64.f32		2				1			0	161	0	0
64	add.s32	2	2				1			2	65	1	0
65	setp.gt.s32	2	6+2				2			0	70	0	0
66	mov.u32	2	2				1			2	67	1	0
67	sub.s32	2	6+1				1			7	68	1	0
68	add.s32	2	6+1				1			7	69	1	0
69	add.s32	2	6+1				1			7	70	1	0
70	selp.b32	2	6+2				2			8	71	1	0
71	setp.lt.s32	2	4				2			0	73	0	0
72	add.s32	2	2				1			2	73	1	0
73	selp.b32	2	6+2				2			8	74	1	0
74	mul.wide.s32	2	6+1				1			0	76	0	0
75	mov.u64	2	2				1			2	76	1	0
76	add.s64	2	6+1				1			0	78	0	0
77	mul.wide.s32	2	2				1			2	78	1	0
78	add.s64	2	6+1				1			0	139	0	0
79	cvt.f64.f32	2	12				32			0	144	0	0
80	add.s32	2	2				1			2	81	1	0
81	setp.gt.s32	2	6+2				2			0	85	0	0
82	sub.s32	2	2				1			2	83	1	0
83	add.s32	2	6+1				1			7	84	1	0
84	add.s32	2	6+1				1			7	85	1	0
85	selp.b32	2	6+2				2			8	86	1	0
86	setp.lt.s32	2	6+2				2			0	88	0	0
87	add.s32	2	2				1			2	88	1	0
88	selp.b32	2	6+2				2			8	89	1	0
89	mul.wide.s32	2	6+1				1			0	92	0	0
90	mul.wide.s32	2	2				1			2	91	1	0
91	add.s64	2	6+1				1			7	92	1	0
92	add.s64	2	6+1				1			0	151	0	0
93	cvt.f64.f32	2	12				32			0	155	0	0
94	mov.u32	2	2				1			2	95	1	0
95	sub.s32	2	6+1				1			0	169	0	0
96	mov.u32	2	2				1			1	98	0	0
--	BB0_4:									--	--	--	--
97	mov.u32	2	6+1				1			7	98	1	0
98	sub.s32	2	6+1				1			7	99	1	0
99	setp.le.s32	2	6+2				2			0	102	0	0
100	add.s32	2	2				1			2	101	1	0
101	setp.ge.s32	2	6+2				2			8	102	1	0
102	and.pred	2	6+1				1			0	104	0	0
103	mov.u16	2	2				1			0	171	0	0
104	@!%p15 bra BB0_7;	2	4				2			1	0	0	0
105	bra.uni BB0_5;	2	2				1			0	0	0	0
--	BB0_5:									--	--	--	--
106	setp.gt.s32	2	4				2			0	108	0	0
107	setp.lt.s32	2	4				2			4	108	1	0
108	or.pred	2	6+1				1			0	115	0	0
109	shr.s32	2	4				2			0	112	0	0
110	setp.gt.s32	2	4				2			0	114	0	0
111	neg.s32	2	4				2			4	112	1	0
112	and.b32	2	6+1				1			7	113	1	0
113	setp.lt.s32	2	6+2				2			8	114	1	0
114	or.pred	2	6+1				1			7	115	1	0
115	or.pred	2	6+1				1			0	122	0	0
116	shr.s32	2	4				2			0	119	0	0
117	setp.gt.s32	2	4				2			0	121	0	0
118	neg.s32	2	4				2			4	119	1	0
119	and.b32	2	6+1				1			7	120	1	0
120	setp.lt.s32	2	6+2				2			8	121	1	0
121	or.pred	2	6+1				1			7	122	1	0
122	or.pred	2	6+1				1			7	123	1	0
123	@%p28 bra BB0_7;	2	6+2				2			1	0	0	0
124	add.s64	2	2				1			2	125	1	1
125	ld.shared.f32	8				6+4	4			10	126	1	1
126	cvt.f64.f32	2	12				32			0	142	0	0
127	mov.u64	2	2				1			2	128	1	0
128	add.s64	2	6+1				1			7	129	1	0
129	add.s64	2	6+1				1			7	130	1	1
130	ld.shared.f32	8				6+4	4			10	131	1	1
131	cvt.f64.f32	2	12				32			0	144	0	0
132	setp.gt.s32	2	4				2			0	134	0	0
133	add.s32	2	2				1			2	134	1	0
134	selp.b32	2	6+2				2			8	135	1	0

135	mul.wide.s32	2	6+1				1			7	136	1	0
136	add.s64	2	6+1				1			7	137	1	0
137	add.s64	2	6+1				1			7	138	1	1
138	ld.shared.f32	8				6+4	4			0	140	0	0
139	ld.shared.f32	8				8	4			8	140	1	1
140	add.f32	2	6+1				1			7	141	1	0
141	cvt.f64.f32		12				32			0	143	0	1
142	add.f64	64		12			32			12	143	1	0
143	sub.f64	64		12			32			12	144	1	0
144	fma.rn.f64	64		63+32			32			0	155	0	1
145	setp.gt.s32		4				2			0	147	0	0
146	add.s32	2	2				1			2	147	1	0
147	selp.b32	2	6+2				2			8	148	1	0
148	mul.wide.s32	2	6+1				1			7	149	1	0
149	add.s64	2	6+1				1			7	150	1	1
150	ld.shared.f32	8				6+4	4			0	152	0	0
151	ld.shared.f32	8				8	4			8	152	1	1
152	add.f32	2	6+1				1			7	153	1	0
153	cvt.f64.f32	2	12				32			12	154	1	0
154	sub.f64	64		12			32			12	155	1	0
155	fma.rn.f64	64		63+32			32			0	160	0	1
156	mov.f32		2				1			2	157	1	0
157	sub.f32	2	6+1				1			7	158	1	0
158	mul.f32	2	6+1				1			7	159	1	0
159	cvt.f64.f32	2	12				32			12	160	1	0
160	add.f64	64		12			32			12	161	1	0
161	fma.rn.f64	64		63+32			32			97	162	1	1
162	cvt.rn.f32.f64	2	6+4				4			0	166	0	0
163	mov.u64	2	2				1			2	164	1	0
164	add.s64	2	6+1				1			7	165	1	0
165	add.s64	2	6+1				1			7	166	1	1
166	st.shared.f32	8				20+4	4			0	0	0	1
167	mov.u16		2				1			0	171	0	0
--	BB0_7:									--	--	--	--
168	bar.sync	2							118	1	0	0	0
169	setp.eq.s32	2	6+2				2			8	170	1	0
170	@%p31 bra BB0_11;	2	6+2				2			1	0	0	0
171	setp.eq.s16	2	6+2				2			8	172	1	0
172	@%p32 bra BB0_10;	2	6+2				2			1	0	0	0
173	mov.u64	2	2				1			2	174	1	0
174	add.s64	2	6+1				1			7	175	1	0
175	add.s64	2	6+1				1			7	176	1	1
176	ld.shared.f32	8				6+4	4			0	178	0	1
177	add.s64		2				1			2	178	1	1
178	st.shared.f32	8				20+4	4			0	0	0	0
--	BB0_10:									--	--	--	--
179	bar.sync	2							118	1	0	0	0
180	add.s32	2	6+1				1			0	0	0	0
181	setp.lt.s32	2	4				2			4	182	1	0
182	@%p33 bra BB0_4;	2	6+2				2			1	0	0	0
--	BB0_11:									--	--	--	--
183	and.b16	2	2				1			2	184	1	0
184	setp.eq.s16	2	6+2				2			8	185	1	0
185	@%p34 bra BB0_13;	2	6+2				2			1	0	0	0
186	mul.wide.s32	2	2				1			0	188	0	0
187	mov.u64	2	2				1			2	188	1	0
188	add.s64	2	6+1				1			0	190	0	0
189	mul.wide.s32	2	2				1			2	190	1	0
190	add.s64	2	6+1				1			7	191	1	1
191	ld.shared.f32	8				6+4	4			0	196	0	1
192	mad.lo.s32		2				1			0	194	0	0
193	cvta.to.global.u64	2	8				4			0	195	0	0
194	mul.wide.s32	2	2				1			2	195	1	0
195	add.s64	2	6+1				1			7	196	1	1
196	st.global.f32	8					--	1576		1	0	0	0
--	BB0_13:												
--	ret;												

Table 8-3, Level 1 and Level 2 supersteps for Hotspot on GTX 1070 (s_1=4, s_2=119)

Level1:	Superstep#: <start ~ end, comp, comm, ovh>, Itteration
	1: <1~96,769,3152,118>, 1 2: <97~170,1312,0,118>, 2
	3: <171~182,115,0,118>, 1 4: <183~196,87,1576,0>, 1

Level2: Superstep#: <start ~ end, comp, comm, ovh>

```
1: <1~16,41,0,0>        31: <65~66,12,0,0>      61: <115~118,21,0,0>    91: <157~157,9,0,0>
2: <17~18,11,0,0>       32: <67~67,9,0,0>       62: <119~119,9,0,0>     92: <158~158,9,0,0>
3: <19~19,9,0,0>        33: <68~68,9,0,0>       63: <120~120,10,0,0>    93: <159~159,14,0,0>
4: <20~21,11,0,0>       34: <69~69,9,0,0>       64: <121~121,9,0,0>     94: <160~160,76,0,0>
5: <22~23,11,0,0>       35: <70~70,10,0,0>      65: <122~122,9,0,0>     95: <161~161,159,0,0>
6: <24~25,11,0,0>       36: <71~72,8,0,0>       66: <123~123,10,0,0>    96: <162~163,14,0,0>
7: <26~28,15,0,0>       37: <73~73,10,0,0>      67: <124~124,4,0,0>     97: <164~164,9,0,0>
8: <29~29,10,0,0>       38: <74~75,11,0,0>      68: <125~125,18,0,0>    98: <165~165,9,0,0>
9: <30~32,15,0,0>       39: <76~77,11,0,0>      69: <126~127,16,0,0>    99: <166~168,32,0,118>
10: <33~33,10,0,0>      40: <78~80,23,0,0>      70: <128~128,9,0,0>     100: <169~169,10,0,0>
11: <34~34,9,0,0>       41: <81~82,12,0,0>      71: <129~129,9,0,0>     101: <170~170,10,0,0>
12: <35~35,9,0,0>       42: <83~83,9,0,0>       72: <130~130,18,0,0>    102: <171~171,10,0,0>
13: <36~36,10,0,0>      43: <84~84,9,0,0>       73: <131~133,20,0,0>    103: <172~172,10,0,0>
14: <37~38,11,0,0>      44: <85~85,10,0,0>      74: <134~134,10,0,0>    104: <173~173,4,0,0>
15: <39~40,17,0,0>      45: <86~87,12,0,0>      75: <135~135,9,0,0>     105: <174~174,9,0,0>
16: <41~41,9,0,0>       46: <88~88,10,0,0>      76: <136~136,9,0,0>     106: <175~175,9,0,0>
17: <42~43,10,1576,0>   47: <89~90,11,0,0>      77: <137~137,9,0,0>     107: <176~177,18,0,0>
18: <44~44,4,0,0>       48: <91~91,9,0,0>       78: <138~139,26,0,0>    108: <178~179,32,0,118>
19: <45~46,11,0,0>      49: <92~94,23,0,0>      79: <140~140,9,0,0>     109: <180~181,13,0,0>
20: <47~47,9,0,0>       50: <95~96,11,0,0>      80: <141~142,64,0,0>    110: <182~182,10,0,0>
21: <48~49,32,0,0>      51: <97~97,9,0,0>       81: <143~143,76,0,0>    111: <183~183,4,0,0>
22: <50~50,9,0,0>       52: <98~98,9,0,0>       82: <144~146,159,0,0>   112: <184~184,10,0,0>
23: <51~52,10,1576,0>   53: <99~100,12,0,0>     83: <147~147,10,0,0>    113: <185~185,10,0,0>
24: <53~53,9,0,0>       54: <101~101,10,0,0>    84: <148~148,9,0,0>     114: <186~187,6,0,0>
25: <54~54,9,0,0>       55: <102~104,15,0,0>    85: <149~149,9,0,0>     115: <188~189,11,0,0>
26: <55~56,32,0,118>    56: <105~107,12,0,0>    86: <150~151,26,0,0>    116: <190~190,9,0,0>
27: <57~57,10,0,0>      57: <108~111,21,0,0>    87: <152~152,9,0,0>     117: <191~194,20,0,0>
28: <58~58,10,0,0>      58: <112~112,9,0,0>     88: <153~153,14,0,0>    118: <195~195,9,0,0>
29: <59~59,145,0,0>     59: <113~113,10,0,0>    89: <154~154,76,0,0>    119: <196~196,8,1576,0>
30: <60~64,32,0,0>      60: <114~114,9,0,0>     90: <155~156,159,0,0>
```

$parallel\_comp\ (eq.10) = 1 \times 769 + 2 \times 1312 + 1 \times 115 + 1 \times 87 = 3595$

$block\_bar\_ovh\ (eq.11) = 1 \times 118 + 2 \times 118 + 1 \times 118 + 1 \times 0 = 472$

$block\_comm\ (eq.12) = 1 \times 3152 + 2 \times 0 + 1 \times 0 + 1 \times 1576 = 4728$

$block\_comm_{\Delta} = 3152$

$COMP\ (eq.5) = 2 \times 10 + 3595 = 3615$

$warp\_comp = COMP \div w = 3615 \div 2 = 1808$

$warp\_comm_{\Delta} = 1576$

$$warps\_need\ (eq.15) = 4 \times \left(\left\lceil \frac{1576 \times 265}{1808 \times (3-1)} \right\rceil + 1\right) = 468$$

$$nonoverlapped\ (eq.17) = \min\left\{\frac{4728}{2}, 394 + \frac{3152}{2} \times \max\left(0{,}1 - \frac{2 \times 4}{468}\right)\right\} = 1944$$

$block\_exec\_cycle\ (eq.4) = 358 + 472 + 1944 + 3615 = 6389$

$comp\ (eq.19) = 4087$

$novlp\ (eq.20) = 1944$

$$\rho\ (eq.21) = max\left\{1, min\left\{\left\lfloor \frac{2048}{256} \right\rfloor, \left\lfloor \frac{65536}{256 \times 34} \right\rfloor, \left\lfloor \frac{48\ KB}{3072\ B} \right\rfloor\right\}\right\} = 7$$

$$\mathrm{K}\ (eq.22) = \frac{1849}{15 \times 7} = 17.609$$

$$\tau\ (eq.23) = \left\lceil \frac{1944}{4087} \right\rceil + 1 = 2$$

$\mu = 3.49$

$$\rho \geq \tau : kernel\_exec\_cycle\ (eq.24) = 358 + \frac{1849}{15} \times \frac{4087}{3.49} + \frac{1944}{2} = 145683$$

$mesured\_kernel\_exec\_cycle = 150816$

$$error\ \% = \frac{|150816 - 145683|}{150816} \times 100 = 3.40\ \%$$

9. Details of results for KNN on GTX 1070

$d = 3$ $n_b = 168$ $g_0 = 19$ $\max\_thread\_per\_sm = 2048$

$n_c = 128$ $n_t = 256$ $g_1 = 207$ $reg\_per\_thread = 9$

$n_{SM} = 15$ $h = 28$ $g_2 = 168$ $shmem\_size = 48\ KB$

$n_{ws} = 4$ $l = 28$ $mem\_lat = G_2 = 394$ $shmem\_per\_block = 0\ B$

$n_{du} = 8$ $l_c = 26$ $warp\_lnch\_ovh = 10$ $n_{reg} = 65536$

$n_{fu} = 4$ $l_m = 2$ $block\_lnch\_ovh = 358$

$warp\_size = 32$ $l_b = 0$ $issue\_cycle\ = 1$

$$w\ (eq.\ 6) = \left\lceil \frac{256}{32 \times 4} \right\rceil = 2$$

Table 9-1, Latency, FUs, Throughput, u, d and p for KNN on GTX 1070

Inst#	Instructions	FU	n^i_{fu}	T_i	n^i_{ma}	Latency		u	d	p
						comp	comm			
1	ld.param.u64 %rd1, [Kernel_param_0];	SPs	32	32		6		15	0	0
2	ld.param.u64 %rd2, [Kernel_param_1];	SPs	32	32		6		16	0	0
3	ld.param.u32 %r2, [Kernel_param_2];	SPs	32	32		6		13	0	0
4	ld.param.f32 %f1, [Kernel_param_3];	SPs	32	32		6		22	0	0
5	ld.param.f32 %f2, [Kernel_param_4];	SPs	32	32		6		24	0	0
6	mov.u32 %r3, %ctaid.y;	SPs	32	32		29		9	0	0
7	mov.u32 %r4, %nctaid.x;	SPs	32	32		6		9	0	0
8	mov.u32 %r5, %ctaid.x;	SPs	32	32		29		9	1	0
9	mad.lo.s32 %r6, %r3, %r4, %r5;	SPs	32	32		6		12	0	0
10	mov.u32 %r7, %ntid.x;	SPs	32	32		6		12	0	0
11	mov.u32 %r8, %tid.x;	SPs	32	32		29		12	0	0
12	mad.lo.s32 %r1, %r6, %r7, %r8;	SPs	32	32		6		13	1	0
13	setp.ge.s32 %p1, %r1, %r2;	SPs	32	16		6		14	1	0
14	@%p1 bra BB0_2;	SPs	32	16		6		0	0	0
15	cvta.to.global.u64 %rd3, %rd1;	SPs	32	8		6		20	0	0
16	cvta.to.global.u64 %rd4, %rd2;	SPs	32	8		6		18	0	0
17	mul.wide.s32 %rd5, %r1, 4;	SPs	32	32		6		18	1	0
18	add.s64 %rd6, %rd4, %rd5;	SPs	32	32		6		28	0	0
19	mul.wide.s32 %rd7, %r1, 8;	SPs	32	32		6		20	1	0
20	add.s64 %rd8, %rd3, %rd7;	SPs	32	32		6		21	1	1
21	ld.global.f32 %f3, [%rd8];	LDST	8	8	2		394	22	1	1
22	sub.f32 %f4, %f1, %f3;	SPs	32	32		6		26	0	1
23	ld.global.f32 %f5, [%rd8+4];	LDST	8	8	2	19[1]		24	1	1
24	sub.f32 %f6, %f2, %f5;	SPs	32	32		6		25	1	0
25	mul.f32 %f7, %f6, %f6;	SPs	32	32		6		26	1	0
26	fma.rn.f32 %f8, %f4, %f4, %f7;	SPs	32	32		19		27	1	1
27	sqrt.rn.f32 %f9, %f8;	SFU	8	8		366		28	1	1
28	st.global.f32 [%rd6], %f9;	LDST	8	8	2		394	0	--	--
--	BB0_2:									
--	ret;									

Table 9-2, BLOCK data structure for KNN on GTX 1070

Inst#	Instructions	$comp_i$						$comm_i$	ovh_i	$sync_i$	u	d	p
		Issue	FU_1	FU_2	FU_3	FU_4	$32/T_i$						
1	ld.param.u64	2	6+1				1			0	15	0	0
2	ld.param.u64	2	2				1			0	16	0	0
3	ld.param.u32	2	2				1			0	13	0	0
4	ld.param.f32	2	2				1			0	22	0	0
5	ld.param.f32	2	2				1			0	24	0	0
6	mov.u32	2	2				1			0	9	0	0
7	mov.u32	2	2				1			0	9	0	0
8	mov.u32	2	2				1			2	9	1	0
9	mad.lo.s32	2	6+1				1			0	12	0	0
10	mov.u32	2	2				1			0	12	0	0
11	mov.u32	2	2				1			2	12	0	0
12	mad.lo.s32	2	6+1				1			7	13	1	0
13	setp.ge.s32	2	6+2				2			8	14	1	0
14	@%p1 bra BB0_2;	2	6+2				2			1	0	0	0
15	cvta.to.global.u64	2	4				4			0	20	0	0
16	cvta.to.global.u64	2	4				4			0	18	0	0

[1] L1 cache hit

17	mul.wide.s32	2	2				1			2	18	1	0
18	add.s64	2	6+1				1			0	28	0	0
19	mul.wide.s32	2	2				1			2	20	1	0
20	add.s64	2	6+1				1			7	21	1	1
21	ld.global.f32	8					4	1576		1	22	1	1
22	sub.f32		6+1				1			0	26	0	1
23	ld.global.f32	8				19+12	4			31	24	1	1
24	sub.f32	2	6+1				1			7	25	1	0
25	mul.f32	2	6+1				1			7	26	1	0
26	fma.rn.f32	2	6+1				1			7	27	1	1
27	sqrt.rn.f32	8			366+4		4			370	28	1	1
28	st.global.f32	8					4	1576		1	0	--	--
--	BB0_2:												
--	ret;												

Table 9-3, Level 1 and Level 2 supersteps for KNN on GTX 1070 (s_1=2, s_2=15)

Level1:	Superstep#: <start ~ end, comp, comm, ovh>, Itteration			
	1: <1~14,65,0,0>, 1	2: <15~28,484,3152,0>, 2		
Level2:	Superstep#: <start ~ end, comp, comm, ovh>			
1: <1~8,23,0,0>	5: <14~14,10,0,0>	9: <21~21,8,1576,0>	13: <26~26,9,0,0>	
2: <9~11,13,0,0>	6: <15~17,12,0,0>	10: <22~23,31,0,0>	14: <27~27,378,0,0>	
3: <12~12,9,0,0>	7: <18~19,11,0,0>	11: <24~24,9,0,0>	15: <28~28,8,1576,0>	
4: <13~13,10,0,0>	8: <20~20,9,0,0>	12: <25~25,9,0,0>		

$parallel\_comp\ (eq.10) = 1 \times 65 + 1 \times 484 = 549$

$block\_bar\_ovh\ (eq.11) = 0$

$block\_comm\ (eq.12) = 1 \times 0 + 1 \times 3152 = 3152$

$block\_comm_{\Delta} = 1576$

$COMP\ (eq.5) = 2 \times 10 + 549 = 569$

$warp\_comp = COMP \div w = 569 \div 2 = 285$

$warp\_comm_{\Delta} = 788$

$$warps\_need\ (eq.15) = 4 \times \left(\left\lceil \frac{788 \times 26}{285 \times (2-1)} \right\rceil + 1\right) = 292$$

$$nonoverlapped\ (eq.17) = min\left\{\frac{3152}{2}, 394 + \frac{1576}{2} \times max\left(0{,}1 - \frac{2 \times 4}{292}\right)\right\} = 1161$$

$block\_exec\_cycle\ (eq.4) = 358 + 0 + 1161 + 569 = 2088$

$comp\ (eq.19) = 569$

$novlp\ (eq.20) = 1161$

$$\rho\ (eq.21) = max\left\{1, min\left\{\left\lfloor \frac{2048}{256} \right\rfloor, \left\lfloor \frac{65536}{256 \times 0} \right\rfloor, \left\lfloor \frac{48\ KB}{0\ B} \right\rfloor\right\}\right\} = 8$$

$$\mathrm{K}\ (eq.22) = \frac{168}{15 \times 8} = 1.4$$

$$\tau\ (eq.23) = \left\lceil \frac{1161}{569} \right\rceil + 1 = 4$$

$\mu = 3.49$

$$\rho \geq \tau : kernel\_exec\_cycle\ (eq.24) = 358 + \frac{168}{15} \times \frac{569}{3.49} + \frac{1161}{2} = 2765$$

$mesured\_kernel\_exec\_cycle = 2934$

$$error\ \% = \frac{|2934 - 2765|}{2934} \times 100 = 5.76\ \%$$

10. Details of results for MM on GTX 1070

$d = 3$	$n_b = 200$	$g_0 = 19$	$\max\_thread\_per\_sm = 2048$
$n_c = 128$	$n_t = 1024$	$g_1 = 207$	$reg\_per\_thread = 22$
$n_{SM} = 15$	$h = 155$	$g_2 = 168$	$shmem\_size = 48\ KB$
$n_{ws} = 4$	$l = 1163$	$mem\_lat = G_2 = 394$	$shmem\_per\_block = 2048\ B$
$n_{du} = 8$	$l_c = 1122$	$warp\_lnch\_ovh = 10$	$n_{reg} = 65536$
$n_{fu} = 4$	$l_m = 21$	$block\_lnch\_ovh = 358$	
$warp\_size = 32$	$l_b = 20$	$issue\_cycle\ = 1$	

$$w\ (eq.\ 6) = \left\lceil \frac{1024}{32 \times 4} \right\rceil = 8$$

Table 10-1, Latency, FUs, Throughput, u, d and p for MM on GTX 1070

Inst#	Instructions	FU	n^i_{fu}	T_i	n^i_{ma}	Latency (Cycle)		u	d	p
						محاسبات	ارتباطات			
1	ld.param.u64 %rd7, [Kernel_param_0];	SPs	32	32		6		152	0	0
2	ld.param.u64 %rd8, [Kernel_param_1];	SPs	32	32		6		15	0	0
3	ld.param.u64 %rd9, [Kernel_param_2];	SPs	32	32		6		14	0	0
4	ld.param.u32 %r12, [Kernel_param_3];	SPs	32	32		6		7	0	0
5	ld.param.u32 %r13, [Kernel_param_4];	SPs	32	32		6		10	0	0
6	mov.u32 %r14, %ctaid.y;	SPs	32	32		29		7	1	0
7	mul.lo.s32 %r15, %r12, %r14;	SPs	32	32		6		8	1	0
8	shl.b32 %r30, %r15, 5;	SPs	32	16		6		17	0	0
9	add.s32 %r2, %r12, -1;	SPs	32	32		6		11	0	0
10	shl.b32 %r3, %r13, 5;	SPs	32	16		6		141	0	0
11	setp.lt.s32 %p1, %r2, 0;	SPs	32	16		6		13	0	0
12	mov.f32 %f103, 0f00000000;	SPs	32	32		6		32	0	0
13	@%p1 bra BB0_3;	SPs	32	16		6		0	0	0
14	cvta.to.global.u64 %rd1, %rd9;	SPs	32	8		6		40	0	0
15	cvta.to.global.u64 %rd2, %rd8;	SPs	32	8		6		35	0	0
16	mov.u32 %r16, %tid.x;	SPs	32	32		29		21	0	0
17	add.s32 %r4, %r2, %r30;	SPs	32	32		6		143	0	0
18	mov.u32 %r17, %ctaid.x;	SPs	32	32		29		19	1	0
19	shl.b32 %r29, %r17, 5;	SPs	32	16		6		38	0	0
20	mov.u32 %r18, %tid.y;	SPs	32	32		29		21	1	0
21	mad.lo.s32 %r6, %r18, %r12, %r16;	SPs	32	32		6		33	0	0
22	mul.wide.s32 %rd10, %r18, 128;	SPs	32	32		6		24	0	0
23	mov.u64 %rd11, Kernel $__cuda_...;	SPs	32	32		6		24	1	0
24	add.s64 %rd5, %rd11, %rd10;	SPs	32	32		6		26	0	0
25	mul.wide.s32 %rd12, %r16, 4;	SPs	32	32		6		26	1	0
26	add.s64 %rd3, %rd5, %rd12;	SPs	32	32		6		37	0	0
27	mad.lo.s32 %r7, %r18, %r13, %r16;	SPs	32	32		6		38	0	0
28	mov.u64 %rd13, Kernel $__cuda_...;	SPs	32	32		6		29	1	0
29	add.s64 %rd14, %rd13, %rd10;	SPs	32	32		6		30	1	0
30	add.s64 %rd4, %rd14, %rd12;	SPs	32	32		6		42	0	0
31	add.s64 %rd6, %rd13, %rd12;	SPs	32	32		6		44	0	0
32	mov.f32 %f103, 0f00000000;	SPs	32	32		6		46	0	0
--	BB1_2:							--	--	--
33	add.s32 %r19, %r6, %r30;	SPs	32	32		6		34	1	0
34	mul.wide.s32 %rd15, %r19, 4;	SPs	32	32		6		35	1	0
35	add.s64 %rd16, %rd2, %rd15;	SPs	32	32		6		36	1	1
36	ld.global.f32 %f6, [%rd16];	LDST	8	8	1		394	37	1	0
37	st.shared.f32 [%rd3], %f6;	LDST	8	8	1	20		0	0	1
38	add.s32 %r20, %r7, %r29;	SPs	32	32		6		39	1	0
39	mul.wide.s32 %rd17, %r20, 4;	SPs	32	32		6		40	1	0
40	add.s64 %rd18, %rd1, %rd17;	SPs	32	32		6		41	1	1
41	ld.global.f32 %f7, [%rd18];	LDST	8	8	1		394	42	1	0
42	st.shared.f32 [%rd4], %f7;	LDST	8	8	1	20		0	0	0
43	bar.sync 0;		MI			223		0	0	0
44	ld.shared.f32 %f8, [%rd6];	LDST	8	8	1	6		46	0	0
45	ld.shared.f32 %f9, [%rd5];	LDST	8	8	1	6		46	1	1
46	fma.rn.f32 %f10, %f9, %f8, %f103;	SPs	32	32		19		49	0	1
47	ld.shared.f32 %f11, [%rd6+128];	LDST	8	8	1	6		49	0	0
48	ld.shared.f32 %f12, [%rd5+4];	LDST	8	8	1	6		49	1	1
49	fma.rn.f32 %f13, %f12, %f11, %f10;	SPs	32	32		19		52	0	1
50	ld.shared.f32 %f14, [%rd6+256];	LDST	8	8	1	6		52	0	0
51	ld.shared.f32 %f15, [%rd5+8];	LDST	8	8	1	6		52	1	1
52	fma.rn.f32 %f16, %f15, %f14, %f13;	SPs	32	32		19		55	0	1
53	ld.shared.f32 %f17, [%rd6+384];	LDST	8	8	1	6		55	0	0

54	ld.shared.f32	%f18, [%rd5+12];	LDST	8	8	1	6		55	1	1
55	fma.rn.f32	%f19, %f18, %f17, %f16;	SPs	32	32		19		58	0	1
56	ld.shared.f32	%f20, [%rd6+512];	LDST	8	8	1	6		58	0	0
57	ld.shared.f32	%f21, [%rd5+16];	LDST	8	8	1	6		58	1	1
58	fma.rn.f32	%f22, %f21, %f20, %f19;	SPs	32	32		19		61	0	1
59	ld.shared.f32	%f23, [%rd6+640];	LDST	8	8	1	6		61	0	0
60	ld.shared.f32	%f24, [%rd5+20];	LDST	8	8	1	6		61	1	1
61	fma.rn.f32	%f25, %f24, %f23, %f22;	SPs	32	32		19		64	0	1
62	ld.shared.f32	%f26, [%rd6+768];	LDST	8	8	1	6		64	0	0
63	ld.shared.f32	%f27, [%rd5+24];	LDST	8	8	1	6		64	1	1
64	fma.rn.f32	%f28, %f27, %f26, %f25;	SPs	32	32		19		67	0	1
65	ld.shared.f32	%f29, [%rd6+896];	LDST	8	8	1	6		67	0	0
66	ld.shared.f32	%f30, [%rd5+28];	LDST	8	8	1	6		67	1	1
67	fma.rn.f32	%f31, %f30, %f29, %f28;	SPs	32	32		19		70	0	1
68	ld.shared.f32	%f32, [%rd6+1024];	LDST	8	8	1	6		70	0	0
69	ld.shared.f32	%f33, [%rd5+32];	LDST	8	8	1	6		70	1	1
70	fma.rn.f32	%f34, %f33, %f32, %f31;	SPs	32	32		19		73	0	1
71	ld.shared.f32	%f35, [%rd6+1152];	LDST	8	8	1	6		73	0	0
72	ld.shared.f32	%f36, [%rd5+36];	LDST	8	8	1	6		73	1	1
73	fma.rn.f32	%f37, %f36, %f35, %f34;	SPs	32	32		19		76	0	1
74	ld.shared.f32	%f38, [%rd6+1280];	LDST	8	8	1	6		76	0	0
75	ld.shared.f32	%f39, [%rd5+40];	LDST	8	8	1	6		76	1	1
76	fma.rn.f32	%f40, %f39, %f38, %f37;	SPs	32	32		19		79	0	1
77	ld.shared.f32	%f41, [%rd6+1408];	LDST	8	8	1	6		79	0	0
78	ld.shared.f32	%f42, [%rd5+44];	LDST	8	8	1	6		79	1	1
79	fma.rn.f32	%f43, %f42, %f41, %f40;	SPs	32	32		19		82	0	1
80	ld.shared.f32	%f44, [%rd6+1536];	LDST	8	8	1	6		82	0	0
81	ld.shared.f32	%f45, [%rd5+48];	LDST	8	8	1	6		82	1	1
82	fma.rn.f32	%f46, %f45, %f44, %f43;	SPs	32	32		19		85	0	1
83	ld.shared.f32	%f47, [%rd6+1664];	LDST	8	8	1	6		85	0	0
84	ld.shared.f32	%f48, [%rd5+52];	LDST	8	8	1	6		85	1	1
85	fma.rn.f32	%f49, %f48, %f47, %f46;	SPs	32	32		19		88	0	1
86	ld.shared.f32	%f50, [%rd6+1792];	LDST	8	8	1	6		88	0	0
87	ld.shared.f32	%f51, [%rd5+56];	LDST	8	8	1	6		88	1	1
88	fma.rn.f32	%f52, %f51, %f50, %f49;	SPs	32	32		19		91	0	1
89	ld.shared.f32	%f53, [%rd6+1920];	LDST	8	8	1	6		91	0	0
90	ld.shared.f32	%f54, [%rd5+60];	LDST	8	8	1	6		91	1	1
91	fma.rn.f32	%f55, %f54, %f53, %f52;	SPs	32	32		19		94	0	1
92	ld.shared.f32	%f56, [%rd6+2048];	LDST	8	8	1	6		94	0	0
93	ld.shared.f32	%f57, [%rd5+64];	LDST	8	8	1	6		94	1	1
94	fma.rn.f32	%f58, %f57, %f56, %f55;	SPs	32	32		19		97	0	1
95	ld.shared.f32	%f59, [%rd6+2176];	LDST	8	8	1	6		97	0	0
96	ld.shared.f32	%f60, [%rd5+68];	LDST	8	8	1	6		97	1	1
97	fma.rn.f32	%f61, %f60, %f59, %f58;	SPs	32	32		19		100	0	1
98	ld.shared.f32	%f62, [%rd6+2304];	LDST	8	8	1	6		100	0	0
99	ld.shared.f32	%f63, [%rd5+72];	LDST	8	8	1	6		100	1	1
100	fma.rn.f32	%f64, %f63, %f62, %f61;	SPs	32	32		19		103	0	1
101	ld.shared.f32	%f65, [%rd6+2432];	LDST	8	8	1	6		103	0	0
102	ld.shared.f32	%f66, [%rd5+76];	LDST	8	8	1	6		103	1	1
103	fma.rn.f32	%f67, %f66, %f65, %f64;	SPs	32	32		19		106	0	1
104	ld.shared.f32	%f68, [%rd6+2560];	LDST	8	8	1	6		106	0	0
105	ld.shared.f32	%f69, [%rd5+80];	LDST	8	8	1	6		106	1	1
106	fma.rn.f32	%f70, %f69, %f68, %f67;	SPs	32	32		19		109	0	1
107	ld.shared.f32	%f71, [%rd6+2688];	LDST	8	8	1	6		109	0	0
108	ld.shared.f32	%f72, [%rd5+84];	LDST	8	8	1	6		109	1	1
109	fma.rn.f32	%f73, %f72, %f71, %f70;	SPs	32	32		19		112	0	1
110	ld.shared.f32	%f74, [%rd6+2816];	LDST	8	8	1	6		112	0	0
111	ld.shared.f32	%f75, [%rd5+88];	LDST	8	8	1	6		112	1	1
112	fma.rn.f32	%f76, %f75, %f74, %f73;	SPs	32	32		19		115	0	1
113	ld.shared.f32	%f77, [%rd6+2944];	LDST	8	8	1	6		115	0	0
114	ld.shared.f32	%f78, [%rd5+92];	LDST	8	8	1	6		115	1	1
115	fma.rn.f32	%f79, %f78, %f77, %f76;	SPs	32	32		19		118	0	1
116	ld.shared.f32	%f80, [%rd6+3072];	LDST	8	8	1	6		118	0	0
117	ld.shared.f32	%f81, [%rd5+96];	LDST	8	8	1	6		118	1	1
118	fma.rn.f32	%f82, %f81, %f80, %f79;	SPs	32	32		19		121	0	1
119	ld.shared.f32	%f83, [%rd6+3200];	LDST	8	8	1	6		121	0	0
120	ld.shared.f32	%f84, [%rd5+100];	LDST	8	8	1	6		121	1	1
121	fma.rn.f32	%f85, %f84, %f83, %f82;	SPs	32	32		19		124	0	1
122	ld.shared.f32	%f86, [%rd6+3328];	LDST	8	8	1	6		124	0	0
123	ld.shared.f32	%f87, [%rd5+104];	LDST	8	8	1	6		124	1	1
124	fma.rn.f32	%f88, %f87, %f86, %f85;	SPs	32	32		19		127	0	1
125	ld.shared.f32	%f89, [%rd6+3456];	LDST	8	8	1	6		127	0	0
126	ld.shared.f32	%f90, [%rd5+108];	LDST	8	8	1	6		127	1	1
127	fma.rn.f32	%f91, %f90, %f89, %f88;	SPs	32	32		19		130	0	1

128	ld.shared.f32 %f92, [%rd6+3584];	LDST	8	8	1	6		130	0	0
129	ld.shared.f32 %f93, [%rd5+112];	LDST	8	8	1	6		130	1	1
130	fma.rn.f32 %f94, %f93, %f92, %f91;	SPs	32	32		19		133	0	1
131	ld.shared.f32 %f95, [%rd6+3712];	LDST	8	8	1	6		133	0	0
132	ld.shared.f32 %f96, [%rd5+116];	LDST	8	8	1	6		133	1	1
133	fma.rn.f32 %f97, %f96, %f95, %f94;	SPs	32	32		19		136	0	1
134	ld.shared.f32 %f98, [%rd6+3840];	LDST	8	8	1	6		136	0	0
135	ld.shared.f32 %f99, [%rd5+120];	LDST	8	8	1	6		136	1	1
136	fma.rn.f32 %f100, %f99, %f98, %f97;	SPs	32	32		19		139	0	1
137	ld.shared.f32 %f101, [%rd6+3968];	LDST	8	8	1	6		139	0	0
138	ld.shared.f32 %f102, [%rd5+124];	LDST	8	8	1	6		139	1	1
139	fma.rn.f32 %f103, %f102, %f101, %f100;	SPs	32	32		19		155	0	0
140	bar.sync 0;		MI			223		0	0	0
141	add.s32 %r29, %r29, %r3;	SPs	32	32		6		0	0	0
142	add.s32 %r30, %r30, 32;	SPs	32	32		6		143	1	0
143	setp.le.s32 %p2, %r30, %r4;	SPs	32	16		6		144	1	0
144	@%p2 bra BB1_2;	SPs	32	16		6		0	0	0
--	BB1_3:									
145	mov.u32 %r22, %ctaid.x;	SPs	32	32		29		146	1	0
146	shl.b32 %r23, %r22, 5;	SPs	32	16		6		147	1	0
147	mad.lo.s32 %r24, %r14, %r3, %r23;	SPs	32	32		6		149	0	0
148	mov.u32 %r25, %tid.x;	SPs	32	32		29		149	1	0
149	add.s32 %r26, %r24, %r25;	SPs	32	32		6		151	0	0
150	mov.u32 %r27, %tid.y;	SPs	32	32		29		151	1	0
151	mad.lo.s32 %r28, %r27, %r13, %r26;	SPs	32	32		6		153	0	1
152	cvta.to.global.u64 %rd19, %rd7;	SPs	32	8		6		154	0	1
153	mul.wide.s32 %rd20, %r28, 4;	SPs	32	32		6		154	1	0
154	add.s64 %rd21, %rd19, %rd20;	SPs	32	32		6		155	1	1
155	st.global.f32 [%rd21], %f103;	LDST	8	8	1		394	0	--	--
--	ret;									

Table 10-2, BLOCK data structure for MM on GTX 1070

Inst#	Instructions	$comp_i$						$comm_i$	ovh_i	$sync_i$	u	d	p
		Issue	FU_1	FU_2	FU_3	FU_4	$32/T_i$						
1	ld.param.u64	8	6+7				1			0	152	0	0
2	ld.param.u64	8	8				1			0	15	0	0
3	ld.param.u64	8	8				1			0	14	0	0
4	ld.param.u32	8	8				1			0	7	0	0
5	ld.param.u32	8	8				1			0	10	0	0
6	mov.u32	8	8				1			8	7	1	0
7	mul.lo.s32	8	6+7				1			13	8	1	0
8	shl.b32	8	6+14				2			0	17	0	0
9	add.s32	8	8				1			12	11	0	0
10	shl.b32	8	16				2			0	141	0	0
11	setp.lt.s32	8	16				2			16	13	0	0
12	mov.f32	8	8				1			0	32	0	0
13	@%p1 bra BB0_3;	8	16				2			0	0	0	0
14	cvta.to.global.u64	8	32				4			0	40	0	0
15	cvta.to.global.u64	8	32				4			0	35	0	0
16	mov.u32	8	8				1			0	21	0	0
17	add.s32	8	8				1			0	143	0	0
18	mov.u32	8	8				1			8	19	1	0
19	shl.b32	8	6+14				2			0	38	0	0
20	mov.u32	8	8				1			8	21	1	0
21	mad.lo.s32	8	6+7				1			0	33	0	0
22	mul.wide.s32	8	8				1			0	24	0	0
23	mov.u64	8	8				1			8	24	1	0
24	add.s64	8	6+7				1			0	26	0	0
25	mul.wide.s32	8	8				1			8	26	1	0
26	add.s64	8	6+7				1			0	37	0	0
27	mad.lo.s32	8	8				1			0	38	0	0
28	mov.u64	8	8				1			8	29	1	0
29	add.s64	8	6+7				1			13	30	1	0
30	add.s64	8	6+7				1			0	42	0	0
31	add.s64	8	8				1			0	44	0	0
32	mov.f32	8	8				1			1	46	0	0
--	BB1_2:										--	--	--
33	add.s32	8	8				1			8	34	1	0
34	mul.wide.s32	8	6+7				1			13	35	1	0
35	add.s64	8	6+7				1			13	36	1	1
36	ld.global.f32	32					4	3152		1	37	1	0
37	st.shared.f32	32				20+28	4			0	0	0	1
38	add.s32		6+7				1			13	39	1	0

39	mul.wide.s32	8	6+7				1			13	40	1	0
40	add.s64	8	6+7				1			13	41	1	1
41	ld.global.f32	32					4	3152		1	42	1	0
42	st.shared.f32	32				20+28	4			0	0	0	0
43	bar.sync 0;	8							223	1	0	0	0
44	ld.shared.f32	32				6+28	4			0	46	0	0
45	ld.shared.f32	32				32	4			32	46	1	1
46	fma.rn.f32		19+7				1			0	49	0	1
47	ld.shared.f32	32				6+28	4			0	49	0	0
48	ld.shared.f32	32				32	4			32	49	1	1
49	fma.rn.f32		19+7				1			0	52	0	1
50	ld.shared.f32	32				6+28	4			0	52	0	0
51	ld.shared.f32	32				32	4			32	52	1	1
52	fma.rn.f32		19+7				1			0	55	0	1
53	ld.shared.f32	32				6+28	4			0	55	0	0
54	ld.shared.f32	32				32	4			32	55	1	1
55	fma.rn.f32		19+7				1			0	58	0	1
56	ld.shared.f32	32				6+28	4			0	58	0	0
57	ld.shared.f32	32				32	4			32	58	1	1
58	fma.rn.f32		19+7				1			0	61	0	1
59	ld.shared.f32	32				6+28	4			0	61	0	0
60	ld.shared.f32	32				32	4			32	61	1	1
61	fma.rn.f32		19+7				1			0	64	0	1
62	ld.shared.f32	32				6+28	4			0	64	0	0
63	ld.shared.f32	32				32	4			32	64	1	1
64	fma.rn.f32		19+7				1			0	67	0	1
65	ld.shared.f32	32				6+28	4			0	67	0	0
66	ld.shared.f32	32				32	4			32	67	1	1
67	fma.rn.f32		19+7				1			0	70	0	1
68	ld.shared.f32	32				6+28	4			0	70	0	0
69	ld.shared.f32	32				32	4			32	70	1	1
70	fma.rn.f32		19+7				1			0	73	0	1
71	ld.shared.f32	32				6+28	4			0	73	0	0
72	ld.shared.f32	32				32	4			32	73	1	1
73	fma.rn.f32		19+7				1			0	76	0	1
74	ld.shared.f32	32				6+28	4			0	76	0	0
75	ld.shared.f32	32				32	4			32	76	1	1
76	fma.rn.f32		19+7				1			0	79	0	1
77	ld.shared.f32	32				6+28	4			0	79	0	0
78	ld.shared.f32	32				32	4			32	79	1	1
79	fma.rn.f32		19+7				1			0	82	0	1
80	ld.shared.f32	32				6+28	4			0	82	0	0
81	ld.shared.f32	32				32	4			32	82	1	1
82	fma.rn.f32		19+7				1			0	85	0	1
83	ld.shared.f32	32				6+28	4			0	85	0	0
84	ld.shared.f32	32				32	4			32	85	1	1
85	fma.rn.f32		19+7				1			0	88	0	1
86	ld.shared.f32	32				6+28	4			0	88	0	0
87	ld.shared.f32	32				32	4			32	88	1	1
88	fma.rn.f32		19+7				1			0	91	0	1
89	ld.shared.f32	32				6+28	4			0	91	0	0
90	ld.shared.f32	32				32	4			32	91	1	1
91	fma.rn.f32		19+7				1			0	94	0	1
92	ld.shared.f32	32				6+28	4			0	94	0	0
93	ld.shared.f32	32				32	4			32	94	1	1
94	fma.rn.f32		19+7				1			0	97	0	1
95	ld.shared.f32	32				6+28	4			0	97	0	0
96	ld.shared.f32	32				32	4			32	97	1	1
97	fma.rn.f32		19+7				1			0	100	0	1
98	ld.shared.f32	32				6+28	4			0	100	0	0
99	ld.shared.f32	32				32	4			32	100	1	1
100	fma.rn.f32		19+7				1			0	103	0	1
101	ld.shared.f32	32				6+28	4			0	103	0	0
102	ld.shared.f32	32				32	4			32	103	1	1
103	fma.rn.f32		19+7				1			0	106	0	1
104	ld.shared.f32	32				6+28	4			0	106	0	0
105	ld.shared.f32	32				32	4			32	106	1	1
106	fma.rn.f32		19+7				1			0	109	0	1
107	ld.shared.f32	32				6+28	4			0	109	0	0
108	ld.shared.f32	32				32	4			32	109	1	1
109	fma.rn.f32		19+7				1			0	112	0	1
110	ld.shared.f32	32				6+28	4			0	112	0	0
111	ld.shared.f32	32				32	4			32	112	1	1
112	fma.rn.f32		19+7				1			0	115	0	1

113	ld.shared.f32	32				6+28	4			0	115	0	0
114	ld.shared.f32	32				32	4			32	115	1	1
115	fma.rn.f32		19+7				1			0	118	0	1
116	ld.shared.f32	32				6+28	4			0	118	0	0
117	ld.shared.f32	32				32	4			32	118	1	1
118	fma.rn.f32		19+7				1			0	121	0	1
119	ld.shared.f32	32				6+28	4			0	121	0	0
120	ld.shared.f32	32				32	4			32	121	1	1
121	fma.rn.f32		19+7				1			0	124	0	1
122	ld.shared.f32	32				6+28	4			0	124	0	0
123	ld.shared.f32	32				32	4			32	124	1	1
124	fma.rn.f32		19+7				1			0	127	0	1
125	ld.shared.f32	32				6+28	4			0	127	0	0
126	ld.shared.f32	32				32	4			32	127	1	1
127	fma.rn.f32		19+7				1			0	130	0	1
128	ld.shared.f32	32				6+28	4			0	130	0	0
129	ld.shared.f32	32				32	4			32	130	1	1
130	fma.rn.f32		19+7				1			0	133	0	1
131	ld.shared.f32	32				6+28	4			0	133	0	0
132	ld.shared.f32	32				32	4			32	133	1	1
133	fma.rn.f32		19+7				1			0	136	0	1
134	ld.shared.f32	32				6+28	4			0	136	0	0
135	ld.shared.f32	32				32	4			32	136	1	1
136	fma.rn.f32		19+7				1			0	139	0	1
137	ld.shared.f32	32				6+28	4			0	139	0	0
138	ld.shared.f32	32				32	4			32	139	1	1
139	fma.rn.f32	8	19+7				1			0	155	0	0
140	bar.sync 0;								223	1	0	0	0
141	add.s32	8	6+7				1			0	0	0	0
142	add.s32	8	8				1			8	143	1	0
143	setp.le.s32	8	6+7				2			13	144	1	0
144	@%p2 bra BB1_2;	8	6+7				2			1	0	0	0
--	BB1_3:												
145	mov.u32	8	8				1			8	146	1	0
146	shl.b32	8	6+14				2			20	147	1	0
147	mad.lo.s32	8	6+7				1			0	149	0	0
148	mov.u32	8	8				1			8	149	1	0
149	add.s32	8	6+7				1			0	151	0	0
150	mov.u32	8	8				1			8	151	1	0
151	mad.lo.s32		6+7				1			5	153	0	1
152	cvta.to.global.u64	8	8				4			0	154	0	1
153	mul.wide.s32	8	8				1			8	154	1	0
154	add.s64	8	6+7				1			13	155	1	1
155	st.global.f32	32					4	3152		1	0	--	--
--	ret;												

Table 10-3, Level 1 and Level 2 supersteps for MM on GTX 1070 (s_1=3, s_2=64)

Level1: Superstep#: <start ~ end, comp, comm, ovh>, Itteration

1: <1~32,475,0,0>, 1 2: <33~144,2573,6304,446>, 10 3: <145~155,192,3152,0>, 1

Level2: Superstep#: <start ~ end, comp, comm, ovh>

1: <1~6,61,0,0>	17: <39~39,21,0,0>	33: <79~81,66,0,0>	49: <127~129,66,0,0>
2: <7~7,21,0,0>	18: <40~40,21,0,0>	34: <82~84,66,0,0>	50: <130~132,66,0,0>
3: <8~9,36,0,0>	19: <41~41,32,3152,0>	35: <85~87,66,0,0>	51: <133~135,66,0,0>
4: <10~11,40,0,0>	20: <42~43,80,0,223>	36: <88~90,66,0,0>	52: <136~138,66,0,0>
5: <12~18,120,0,0>	21: <44~45,98,0,0>	37: <91~93,66,0,0>	53: <139~140,34,0,223>
6: <19~20,36,0,0>	22: <46~48,66,0,0>	38: <94~96,66,0,0>	54: <141~142,29,0,0>
7: <21~23,37,0,0>	23: <49~51,66,0,0>	39: <97~99,66,0,0>	55: <143~143,21,0,0>
8: <24~25,29,0,0>	24: <52~54,66,0,0>	40: <100~102,66,0,0>	56: <144~144,21,0,0>
9: <26~28,37,0,0>	25: <55~57,66,0,0>	41: <103~105,66,0,0>	57: <145~145,16,0,0>
10: <29~29,21,0,0>	26: <58~60,66,0,0>	42: <106~108,66,0,0>	58: <146~146,28,0,0>
11: <30~32,37,0,0>	27: <61~63,66,0,0>	43: <109~111,66,0,0>	59: <147~148,29,0,0>
12: <33~33,16,0,0>	28: <64~66,66,0,0>	44: <112~114,66,0,0>	60: <149~150,29,0,0>
13: <34~34,21,0,0>	29: <67~69,66,0,0>	45: <115~117,66,0,0>	61: <151~152,21,0,0>
14: <35~35,21,0,0>	30: <70~72,66,0,0>	46: <118~120,66,0,0>	62: <153~153,16,0,0>
15: <36~36,32,3152,0>	31: <73~75,66,0,0>	47: <121~123,66,0,0>	63: <154~154,21,0,0>
16: <37~38,80,0,0>	32: <76~78,66,0,0>	48: <124~126,66,0,0>	64: <155~155,32,3152,0>

$parallel\_comp\ (eq.10) = 1 \times 475 + 10 \times 2573 + 1 \times 192 = 26397$

$block\_bar\_ovh\ (eq.11) = 1 \times 0 + 10 \times 460 + 1 \times 0 = 4460$

$block\_comm\ (eq.12) = 1 \times 0 + 10 \times 6304 + 1 \times 3152 = 66192$

$block\_comm_{\Delta} = 63040$

$COMP\ (eq.5) = 8 \times 10 + 26397 = 26477$

$warp\_comp = 26477 \div 8 = 3310$

$warp\_comm_{\Delta} = 7880$

$$warps\_need\ (eq.15) = 4 \times \left(\left\lceil \frac{7880 \times 1122}{3310 \times (21-1)} \right\rceil + 1\right) = 540$$

$$nonoverlapped\ (eq.17) = \min\left\{\frac{66192}{8}, 394 + \frac{63040}{8} \times \max\left(0{,}1 - \frac{8 \times 4}{540}\right)\right\} = 7808$$

$block\_exec\_cycle\ (eq.4) = 335 + 4460 + 7808 + 26477 = 39080$

$comp\ (eq.19) = 30937$

$novlp\ (eq.20) = 7808$

$$\rho\ (eq.21) = max\left\{1, min\left\{\left\lfloor \frac{2048}{1024} \right\rfloor, \left\lfloor \frac{65536}{1024 \times 22} \right\rfloor, \left\lfloor \frac{48\ KB}{2048\ B} \right\rfloor\right\}\right\} = 2$$

$$\mathrm{K}\ (eq.22) = \frac{200}{15 \times 2} = 6.666$$

$$\tau\ (eq.23) = \left\lceil \frac{7808}{30937} \right\rceil + 1 = 2$$

$\mu = 3.49$

$$\rho \geq \tau : kernel\_exec\_cycle\ (eq.24) = \ 335 + \frac{200}{15} \times \frac{30937}{1.5} + \frac{7808}{2} = 279235$$

$mesured\_kernel\_exec\_cycle = 258574$

$$error\ \% = \frac{|258574 - 279235|}{258574} \times 100 = 7.99\ \%$$