

X


swayam.gov.in

[NPTEL](https://swayam.gov.in/nc_details/NPTEL)

reviewer4@nptel.iitm.ac.in ▾

[NPTEL](https://swayam.gov.in/explorer?ncCode=NPTEL) » [GPU Architectures and Programming \(course\)](#)
[Announcements \(announcements\)](#) [About the Course](https://swayam.gov.in/nd1_noc20_cs41/preview)
[Ask a Question \(forum\)](#) [Progress \(student/home\)](#) [Mentor \(student/mentor\)](#)

Unit 9 - Week 6

Course outline

How does an NPTEL online course work?

Week 0

Week 1

Week 2

Tutorial Videos

Week 3

Week 4

Week 5

Week 6

- Lecture 19 :
Memory Access
Coalescing
(unit?
unit=32&lesson=97)
- Lecture 20 :
Memory Access
Coalescing
(Contd.) (unit?
unit=32&lesson=98)

Assignment 6

The due date for submitting this assignment has passed. **Due on 2020-03-11, 23:59 IST.**
As per our records you have not submitted this assignment.

1)

20 points

- Lecture 21 :
Memory Access
Coalescing
(Contd.) (unit?
unit=32&lesson=99)
- Lecture 22 :
Memory Access
Coalescing
(Contd.) (unit?
unit=32&lesson=100)
- Lecture 23 :
Memory Access
Coalescing
(Contd.) (unit?
unit=32&lesson=101)
- Lecture 24 :
Memory Access
Coalescing
(Contd.) (unit?
unit=32&lesson=102)
- Lecture 25 :
Memory Access
Coalescing
(Contd.) (unit?
unit=32&lesson=103)
- Lecture 26 :
Memory Access
Coalescing
(Contd.) (unit?
unit=32&lesson=104)
- Lecture 27 :
Memory Access
Coalescing
(Contd.) (unit?
unit=32&lesson=105)
- Lecture Material
(unit?
unit=32&lesson=44)
- Quiz :
Assignment 6
(assessment?
name=94)

- Week 6
Programming
Assignment
(/noc20_cs41/subjective?
name=110)
- Week 6
Feedback Form
(unit?
unit=32&lesson=73)

Week 7

Week 8

In a GPU system with memory transaction width of N , a global memory access can coalesce (bring in a single transaction) N consecutive floating point data values. Consider the following code snippet.

```
__global__ void mem_access(float* A)
{
    int tid = threadIdx.x;
    for(int i=1; i<=32; i=i*2)
        A[tid*i]+=2;
}
```

Let the number of threads be 256 ($\text{tid} = 0$ to 255) and the size of the array A be 8192. Assuming a warp size of 16, compute the total number of global memory transactions made in the **for** loop for transaction widths of 16 elements. Assume no caching occurs.

- A. 752
- B. 992
- C. 768
- D. 1024

- A.
- B.
- C.
- D.

No, the answer is incorrect.

Score: 0

Accepted Answers:

A.

2)

20 points

[Week 9](#)[Week 10](#)[Week 11](#)[Lecture Materials](#)[Week 12](#)[Details Solution](#)[Download Videos](#)[Text Transcripts](#)[Live Interactive Session](#)

Consider the following kernel code snippet.

```
#define BDIMX 32
#define BDIMY 32

__global__ void setRowReadCol(int *out)
{
    __shared__ int tile[BDIMY][BDIMX];
    unsigned int row_idx = threadIdx.y * blockDim.x + threadIdx.x;
    unsigned int col_idx = threadIdx.x * blockDim.y + threadIdx.y;
    tile[row_idx] = row_idx;
    __syncthreads();
    out[row_idx] = tile[col_idx];
}
```

The kernel is executed on a GPU architecture where the number of shared memory banks is 16 and the bank width is 4 bytes. The kernel is launched with the following configuration.

```
dim3 block (BDIMX, BDIMY);
```

```
dim3 grid (1,1);
```

Assuming the size of an integer is 4 bytes, match and pair the following.

- | | |
|--|-------|
| i. Number of shared loads per warp request | a. 1 |
| ii. Number of shared stores per warp request | b. 16 |
| iii. Number of global loads per warp request | c. 32 |
| iv. Number of global stores per warp request | |

Select the correct option

- A. i → b, ii→b, iii→a, iv→a
- B. i → b, ii→a, iii→a, iv→a
- C. i → c, ii→c, iii→a, iv→a
- D. i → a, ii→a, iii→b, iv→b

- A.
- B.
- C.
- D.

No, the answer is incorrect.

Score: 0

Accepted Answers:

B.

