Structure Sparsity Extension Introduction

A XuanTie Matrix Extension Subset

Structure Sparsity under AI applications

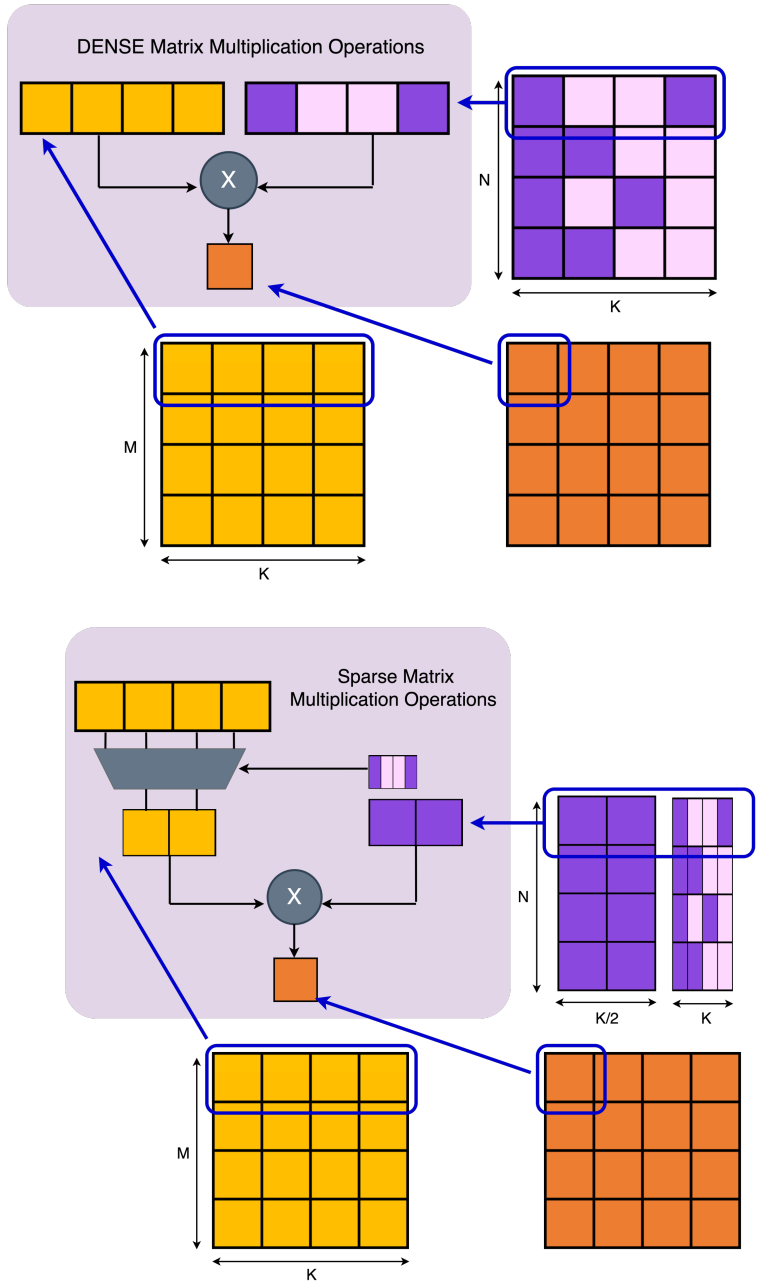
sparse weights (input dimension)
dense activations
dense accumulators

structure format for sparsity

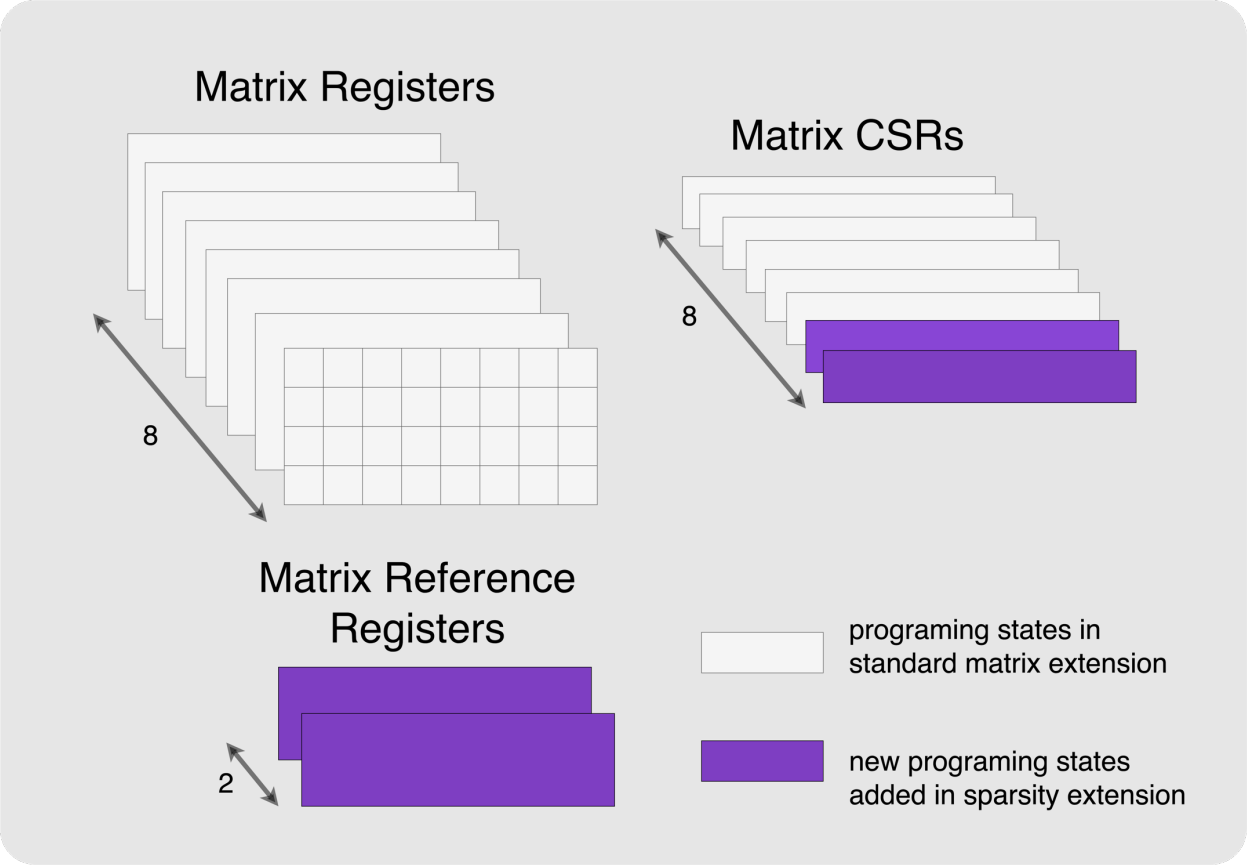
- at most N nonzero in M elements (N:M sparsity)
- do not store 0s in M elements
- reference to index remaining N nonzero values

applicable for

- convolutions
- transformer blocks
- MLPs
- etc.



New Programming States



CSRs

Matrix Register Information register (URO)

sparsityM element numbers in one sparse block

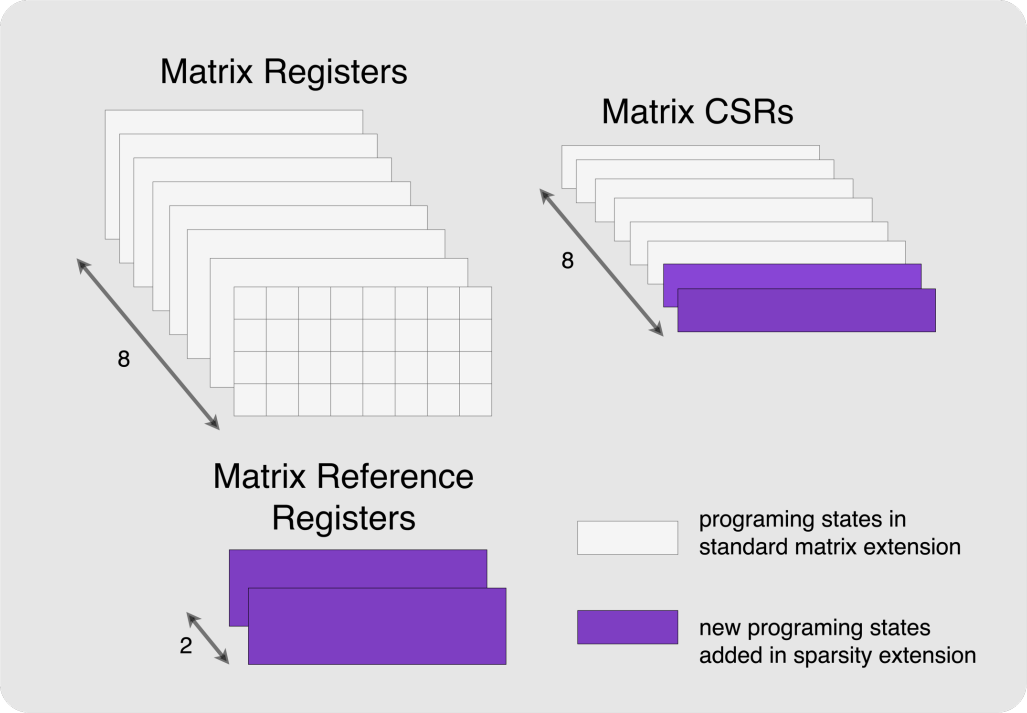
xmrefrlenb matrix reference register length in byte

Matrix Control and Status (URW)

sparsity_ratio current structured sparsity ratio

sparsityM	sparsity_ratio	sparse structure format
4	0	2:4
	1	1:4
8	0	4:8
	1	2:8
	2	1:8

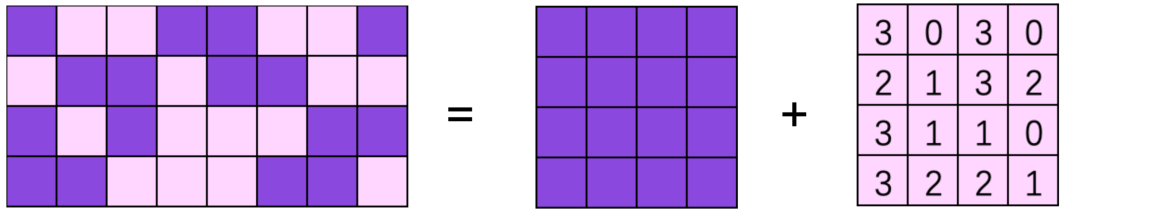
New Programming States



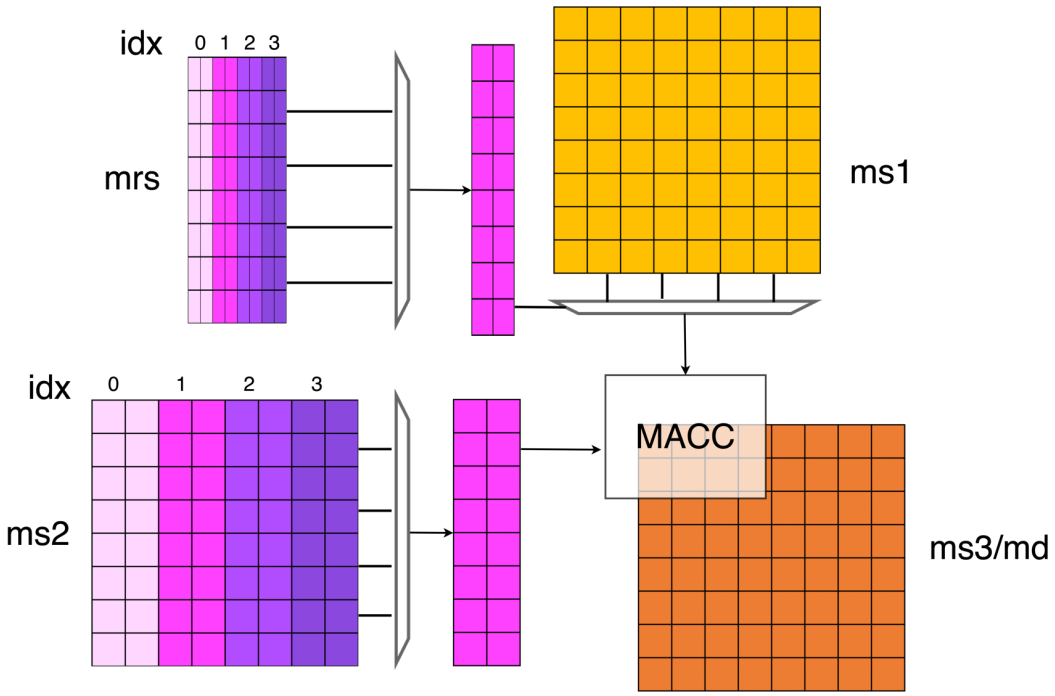
Matrix Reference Register

save references to index remaining N nonzero values

RLEN	Matrix Register Size	sparsityM	Matrix Reference Register Size
128	4 * 128 bit	4	4 * 32 bit
		8	4 * 64 bit
256	8 * 256 bit	4	8 * 64 bit
		8	8 * 128 bit



New Sparsity Instructions



smmaqa.b.<idx> md, ms2, ms1, mrs

integer sparse multiply accumulation(4 widen)

smmaqa.b.<idx> md, ms2, ms1, mrs

float sparse multiply accumulation (non-widen)

sfmmacc.h.<idx> md, ms2, ms1, mrs

float sparse multiply accumulation (double widen)

sfwmmacc.h.<idx> md, ms2, ms1, mrs

For 50% sparsity (2:4 or 4:8)

1/2 matrix register is used to generate one accumulators

For 75% sparsity (1:4 or 2:8)

1/4 matrix register is used to generate one accumulators

New Sparsity Instructions

matrix reference register load/store instructions

mrld mrd, (rs1)， rs2
mrst mrd, (rs1)， rs2

matrix reference register streaming load/store instructions

mrsld mrd, (rs1)， rs2
mrsst mrd, (rs1)， rs2

matrix reference register whole load/store instructions

mrld<1,2> mrd, (rs1)
mrst<1/2>m mrs, (rs1)

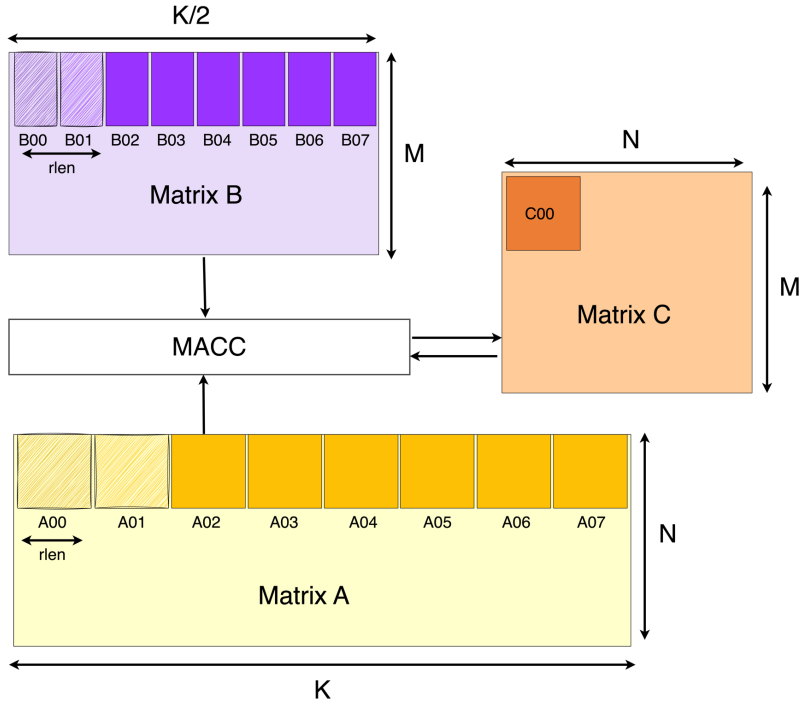
One-to-one correspondence with standard XuanTie Matrix Extension

matrix register load/store instructions
matrix register streaming loa/stored instructions
matrix register whole load/store instructions

SPMM Demos

```
#nt8 sparse matrix multiply

while(not_done){
        mld.b  m3, x2, (x1)              # pre-load A00
        add    x1, x1, sizeK
        mld.b  m2, x4, (x3)              # pre-load B00 B01
        mrld  mr0, x6, (x5)              # pre-load mr00 mr01
        smmaqa.b.0  m4, m2, m3, mr0      # C00 += A00*B00
        add          x3, x3, sizeK
        mld.b  m5, x2, (x1)              # pre-load A01
        add    x1, x1, sizeK
        smmaqa.b.1  m4, m2, m5, mr0      # C00 += A01*B01
        add          x5, x5, mrsizeK
    }
```



Thanks