ASEArch BLAS API

Jonathan Hogg, STFC

September 13, 2012

Contents

1 Introduction

The ASEArch present a subset of the full BLAS functionality that offers alternative or improved versions of those routines offered by the CUBLAS library.
At present only the `_trsv` triangular solve routine offers a significant benefit from usage, though we do also provide source code for `_gemv`.

2 Usage

To use the ASEArch blas, you should `#include "ASEArch_blas.h"` and link against `libASEArch_blas.a`.

3 Enumerated types

```
enum ASEArch_trans {
   ASEARCH_TRANS,
   ASEARCH_NONTRANS,
};

enum ASEArch_uplo {
   ASEARCH_UPR,
   ASEARCH_LWR,
};

enum ASEArch_diag {
   ASEARCH_UNIT,
   ASEARCH_NONUNIT,
};
```

4 Callable subroutines (host)

4.1 `_gemv`: Matrix-vector multiply

```
void ASEArch_dgemv(enum ASEArch_trans trans, int m, int n, double alpha,
      const double a[], int lda, const double x[], int incx, double beta,
      double y[], int incy);
```

Performs the operation $y \leftarrow \beta y + \alpha Ax$ or $y \leftarrow \beta y + \alpha A^T x$.
Note: At present outperformed by CUBLAS. Use CUBLAS instead.

4.2 `_trsv`: Triangular solve

```
void ASEArch_dtrsv(enum ASEArch_uplo uplo, enum ASEArch_trans trans,
      enum ASEArch_diag diag, int n, const double a[], int lda, double x[],
      int incx);
```

Solves one of the following equations for x:

- $Lx = b$
- $L^T x = b$
- $Ux = b$

- $U^T x = b$

The diagonal of L or U may be considered as either Unit or Non-unit.
Note: At present only lower triangular version is implemented.

5 Callable routines (device)

At present device routines are undocumented and subject to change.