230821 updated

Multilevelsenseamp.cpp – dynamic power equation modified

Bus.cpp, Htree.cpp, XYBus.cpp – latency equation modified

230920 updated:

Searchable through: // 230920 updated

(1) New data transfer mode added: synchronous data transfer

In the synchronous data transfer mode, data is transferred in a pipelined fashion during system-level operations, as opposed to buffer-by-buffer like transfer in the existing implementation. In such a system the overall chip latency is primarily determined by the critical delay path along the total data transfer distance. Given the practical limitations on the global bus width often observed in chips, it's probable that the critical delay path responsible for determining chip latency will be from the global buffer to the tile unit. Synchronous data transfer mode reflects such operational characteristics.

Currently, the synchronous data transfer mode is only supported for Htree interconnect topology and on-chip activation function mode, and novel mapping. Such restriction is incorporated into the code.

(2) Modified some minor issues

Updated code files: Buffer.cpp, Tile.cpp, MultilevelSenseAmp.cpp, Subarray.cpp

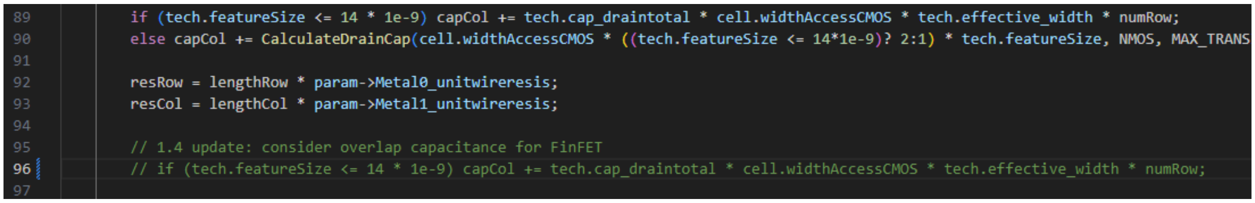
Buffer.cpp – deleted /2 in buffer read/write latency

Tile.cpp - /2 in buffer access counts

MultilevelSenseAmp.cpp, Subarray.cpp – Reference array area estimation was not properly incorporated into the code, so fixed the associated code files.

231030 updated:

Buffer.cpp – typo corrected (cap_draintotal added twice)



```
        if (tech.featureSize <= 14 * 1e-9) capCol += tech.cap_draintotal * cell.widthAccessCMOS * tech.effective_width * numRow;
        else capCol += CalculateDrainCap(cell.widthAccessCMOS * ((tech.featureSize <= 14*1e-9)? 2:1) * tech.featureSize, NMOS, MAX_TRANS

        resRow = lengthRow * param->Metal0_unitwireresis;
        resCol = lengthCol * param->Metal1_unitwireresis;

        // 1.4 update: consider overlap capacitance for FinFET
        // if (tech.featureSize <= 14 * 1e-9) capCol += tech.cap_draintotal * cell.widthAccessCMOS * tech.effective_width * numRow;

```

231229 updated:

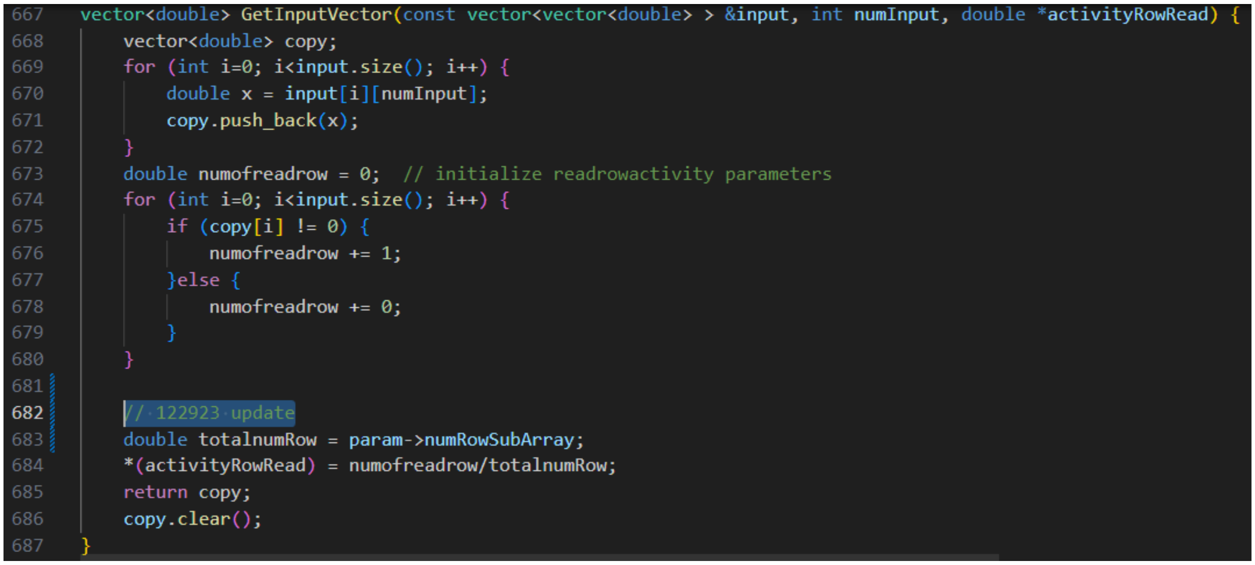
Searchable through: // 122923 update

Updated file: ProcessingUnit.cpp

Updated line:

```
double totalnumRow = param->numRowSubArray;
```

➔ Always turn off unused rows to further save dynamic energy consumption



```
vector<double> GetInputVector(const vector<vector<double> > &input, int numInput, double *activityRowRead) {
    vector<double> copy;
    for (int i=0; i<input.size(); i++) {
        double x = input[i][numInput];
        copy.push_back(x);
    }
    double numofreadrow = 0;  // initialize readrowactivity parameters
    for (int i=0; i<input.size(); i++) {
        if (copy[i] != 0) {
            numofreadrow += 1;
        }else {
            numofreadrow += 0;
        }
    }

    // 122923 update
    double totalnumRow = param->numRowSubArray;
    *(activityRowRead) = numofreadrow/totalnumRow;
    return copy;
    copy.clear();
}
```