Memory expansion module

Rollup team

August 2022

Contents

1 Memory expansion module

1.1 Introduction

The **memory expansion module** is called with every RAM instruction. Its main goal to compute the MEMORY_EXPANSION_GAS_COST (i.e. EXP_GAS) associated with memory expansion triggered by the current instruction and to recognize grossly out of bounds memory operations. If we're being precise, this module to *verifies* the claimed EXP_GAS gas which the current module imports from the central trace as ⟨EXP_GAS⟩. Some instructions (mainly `MSIZE`) don't modify memory size but need to be taken into account regardless so that the memory size MEM_SIZE found in the central trace (and which the current module imports as ⟨MEM_SIZE⟩) is indeed right.

The reason why this is a separate module is that its internal clock (its "heartbeat") is distinct from both that of the RAM pre-processor and the RAM data processor.

There are four broad categories of memory expansions. The zeroth category is for memory instruction that don't have associated memory expansion cost, e.g. `MSIZE`. The first category's expansion cost depends purely on an *offset* as the *size* parameter is known in advance (32 or 1 depending on the instruction):

1. `MLOAD`;
2. `MSTORE`;
3. `MSTORE8`.

The second category computes memory expansion in terms on a a single *offset* and *size*:

1. CREATE;
2. CREATE2;
3. RETURN;
4. REVERT;
5. LOG0-LOG4;
6. SHA3;
7. CODECOPY;
8. EXTCODECOPY;
9. CALLDATACOPY;
10. RETURNDATACOPY.

The third an final category computes memory expansion in terms on two *offsets* and a *sizes*: that of the offset and size defining the next subcontext's calldata

1. CALL;
2. CALLCODE;
3. STATICCALL;
4. DELEGATECALL.

The present module will deal with memory expansion uniformly. It therefore imports from the EVM module two maximal offset columns containing, in the first category and given that $\mathsf{OFFSET}_1^{\mathsf{hi}} = 0$,

1. MLOAD and MSTORE: $\mathsf{MAX\_OFFSET}_1 = \mathsf{OFFSET}_1^{\mathsf{lo}} + 31$, $\mathsf{MAX\_OFFSET}_2 = 0$;
2. MSTORE8: $\mathsf{MAX\_OFFSET}_1 = \mathsf{OFFSET}_1^{\mathsf{lo}}$, $\mathsf{MAX\_OFFSET}_2 = 0$;

for the second category, given that $\mathsf{OFFSET}_1^{\mathsf{hi}} = \mathsf{SIZE}_1^{\mathsf{hi}} = 0$ and $\mathsf{SIZE}_1^{\mathsf{lo}} \neq 0$

$$\mathsf{MAX\_OFFSET}_1 = \mathsf{OFFSET}_1^{\mathsf{lo}} + (\mathsf{SIZE}_1^{\mathsf{lo}} - 1) \quad \text{and} \quad \mathsf{MAX\_OFFSET}_2 = 0,$$

for the third category, given that $\mathsf{OFFSET}_1^{\mathsf{hi}} = \mathsf{SIZE}_1^{\mathsf{hi}} = 0$ and $\mathsf{OFFSET}_2^{\mathsf{hi}} = \mathsf{SIZE}_2^{\mathsf{hi}} = 0$ and $\mathsf{SIZE}_1^{\mathsf{lo}} \neq 0$ or $\mathsf{SIZE}_2^{\mathsf{lo}} \neq 0$:

$$\mathsf{MAX\_OFFSET}_1 = \mathsf{OFFSET}_1^{\mathsf{lo}} + (\mathsf{SIZE}_1^{\mathsf{lo}} - 1) \quad \text{and} \quad \mathsf{MAX\_OFFSET}_2 = \mathsf{OFFSET}_2^{\mathsf{lo}} + (\mathsf{SIZE}_2^{\mathsf{lo}} - 1)$$

For memory instructions that don't (because size is 0) or can't (e.g. MSIZE) incur memory expansion we set $\mathsf{MAX\_OFFSET}_1 = \mathsf{MAX\_OFFSET}_2 = 0$.

1.2 Columns

The following columns determine the heartbeat of the module:

1. $\langle$MEMORY_EXPANSION_STAMP$\rangle$: imported column; abbreviated to $\langle$EXP_STAMP$\rangle$; **TODO: should we just reuse RAM_STAMP instead? We certainly could . . . and probably should.**
2. OFFSET_OUT_OF_BOUNDS: counter-constant binary column indicating whether *both* maximal offsets fit into 3 bytes or not; abbreviated to OOOB;
3. COUNTER: counter column; resets to 0 with every new instruction; counts to either from 0 to 2 or fro 0 to 16 depending on whether maximal offsets fit into 3 bytes ($\mathsf{OOOB} = 0$) or not ($\mathsf{OOOB} = 1$); abbreviated to CT;

The main execution trace constructs maximal offsets as sums of two 16 byte integers. Their sum may thus overflow 16 bytes; computing their byte decomposition may thus require 17 bytes. This explains CT's threshold at 16 rather than the customary 15 found in other modules. *All imported columns are counter-constant.* Furthermore imported columns are surrounded by $\langle \cdots \rangle$.

4. $\langle \textsf{CN} \rangle$: imported column; contains the execution context number currently executing the potentially memory expanding instruction;

5. $\langle \textsf{MAX\_OFFSET}^1 \rangle$ and $\langle \textsf{MAX\_OFFSET}^2 \rangle$: imported columns; contain the greatest byte offset covered (i.e. touched if not modified) by the current instruction;

6. $\textsf{BYTE}^1$ and $\textsf{BYTE}^2$: byte columns;

7. $\textsf{ACC}^1$ and $\textsf{ACC}^2$: accumulator columns;

If maximal offsets aren't grossly out of bound (i.e. if $\textsf{OOOB} = 0$) then $\textsf{BYTE}^1$ and $\textsf{BYTE}^2$ contain the byte decomposition of the 3 byte integers $\langle \textsf{MAX\_OFFSET}^1 \rangle$ and $\langle \textsf{MAX\_OFFSET}^2 \rangle$; otherwise (i.e. if $\textsf{OOOB} = 0$) $\textsf{BYTE}^1$ contains the byte decomposition of $\langle \textsf{MAX\_OFFSET}^1 \rangle - 256^3$ or of $\langle \textsf{MAX\_OFFSET}^2 \rangle - 256^3$: one of which is an integer that fits into 17 bytes or less.

8. $\langle \textsf{MEM\_SIZE} \rangle$: imported (and thus counter-constant) column; contains the size of active memory in bytes (counting continuously from position 0) *before* excution of the current instruction; though imported, its value is only justified here;

9. $\textsf{MEM\_SIZE}^\nu$: counter-constant column; *may* contain the size of active memory in bytes (counting continuously from position 0) *after* excution of the current instruction;

10. $\textsf{EXP\_COST}$: counter-constant column; stores the currently greatest value of "$C_{\text{mem}}(a)$"[1]

11. $\textsf{EXP\_COST}^\nu$: counter-constant column; either retains the value $\textsf{EXP\_COST}$ if no memory expansion took place or, if memory expansion *did* occur, stores the updated value of $\textsf{EXP\_COST}$;

12. $\langle \textsf{MEMORY\_EXPANSION\_GAS\_COST} \rangle$: imported column containing the gas expansion cost $\textsf{EXP\_COST}^\nu - \textsf{EXP\_COST}$; the claimed gas expansion cost is verified in the present module; abbreviated to $\textsf{EXP\_GAS}$;

13. $\textsf{COMP}$: binary counter-constant column; equals 1 *iff* $\textsf{MAX\_OFFSET}^1 \geq \textsf{MAX\_OFFSET}^2$; comes into play only for memory expanding instructions involving two offsets;

14. $\Delta\textsf{\_BYTE}$: a byte column; *may* contain the byte decomposition of an adjusted nonnegative difference between $\textsf{MAX\_OFFSET}^1$ and $\textsf{MAX\_OFFSET}^2$;

15. $\Delta\textsf{\_ACC}$: accumulator column; *may* may accumulate the bytes of $\Delta\textsf{\_BYTE}$;

The **adjusted nonnegative difference** in question is

$$\Delta = \begin{cases} \text{IF } \textsf{MAX\_OFFSET}^1 \geq \textsf{MAX\_OFFSET}^2 : & \textsf{MAX\_OFFSET}^1 - \textsf{MAX\_OFFSET}^2 \\ \text{IF } \textsf{MAX\_OFFSET}^1 < \textsf{MAX\_OFFSET}^2 : & \textsf{MAX\_OFFSET}^2 - \textsf{MAX\_OFFSET}^1 - 1 \end{cases}$$

i.e. we use the same idea as in the word comparison module.

16. $\textsf{MAX\_OFFSET}^{1,2}$: a counter-constant column that equals $\max \left\{ \langle \textsf{MAX\_OFFSET}^1 \rangle, \langle \textsf{MAX\_OFFSET}^2 \rangle \right\}$ *when both are in bounds*;

17. $\textsf{EXP\_FLAG}$: counter-constant binary flag; indicates whether memory expansion occurred;

18. $\textsf{EXP\_BYTE}$: a byte column; *may* contain the byte decomposition of an adjusted difference between $\textsf{MAX\_OFFSET}^{1,2}$ and $\textsf{MEM\_SIZE}$;

[1] Notation taken from the yellow paper:

$$C_{\text{mem}}(a) = G_{\text{mem}} \cdot a + \left\lfloor \frac{a^2}{512} \right\rfloor$$

where a is the current number of evm words in memory.

19. $\mathsf{EXP\_ACC}$: accumulator column; *may* may accumulate the bytes of $\mathsf{EXP\_BYTE}$;

The columns below are introduced purely for computational purposes.

20. $\mathsf{QUOT\_1}$: counter-constant column which *may* contain the quotient of a certain euclidean division;
21. $\mathsf{QUOT\_1\_BYTE}$: byte column which *may* contain the byte decomposition of $\mathsf{QUOT\_1}$;
22. $\mathsf{QUOT\_1\_ACC}$: *may* accumulate the byte from $\mathsf{QUOT\_1\_BYTE}$;
23. $\mathsf{QUOT\_2}$: counter-constant column which *may* contain the quotient of a certain euclidean division;
24. $\mathsf{QUOT\_2\_BYTE}$: byte column which *may* contain the byte decomposition of $\mathsf{QUOT\_2}$;
25. $\mathsf{QUOT\_2\_ACC}$: *may* accumulate the byte from $\mathsf{QUOT\_2\_BYTE}$;
26. $\mathsf{AUX\_1}$ and $\mathsf{AUX\_2}$: two byte columns;

We provide more details. Consider a counter-cycle representing a computation in the "within bounds case" with memory extension. We need to finish the computation by computing the euclidean division of $\mathsf{MEM\_SIZE}_i^\nu = \mathsf{MAX\_OFFSET}_i^{1,2}$ by 32:

$$\mathsf{MAX\_OFFSET}^{1,2} = 32 \cdot \mathsf{QUOT\_1} + \mathtt{r} \tag{$\clubsuit$}$$

For such a cycle, the three rows of $\mathsf{AUX\_1}$ of that counter-cycle will contain in order: $\cdots$, $\mathtt{r}+(256-32)$, $\mathtt{r}$. The bytehood constraint on that column will establish that $\mathtt{r}$ is indeed in the range $\{0, 1, \dots, 31\}$.

We next need to compute, setting $a = 1 + \mathsf{QUOT\_1}$, the euclidean division of a^2 by 512:

$$\begin{cases} a^2 &= & 512 \cdot \mathsf{QUOT\_2} + \mathtt{r}', & 0 \leq \mathtt{r}' < 512, \\ \mathtt{r}' &= & 256 \cdot \epsilon + \mathsf{b}, & \epsilon \in \{0,1\},\ 0 \leq \mathsf{b} < 256, \\ a^2 &= & 512 \cdot \mathsf{QUOT\_2} + 256 \cdot \epsilon + \mathsf{b} & \end{cases} \tag{$\spadesuit$}$$

Given our bounds on $\mathsf{MAX\_OFFSET}^{1,2} < 256^3 = 2^{24}$, we see that

1. $\mathsf{QUOT\_1}$ can be of the order of magnitude of $\approx 2^{19}$
2. $\mathsf{QUOT\_2}$ can be of the order of magnitude of $\approx 2^{29}$

and so $\mathsf{QUOT\_2}$ fits into 4 bytes (rather than 3). We do this using bytes as follows: Let us write $\mathsf{QUOT\_2} = \sum_{i=0}^{3} 256^i \mathsf{b}_i$. $\mathsf{QUOT\_2\_BYTE}$ will thus be made to contain, in order, b_2, b_1, b_0. The three rows of $\mathsf{AUX\_2}$ will contain in order: ϵ, b_3, b. Note that ϵ is a bit rather than any byte; this is tested in the constraint sytem.

1.3 Offset bounds

RAM is a word addressed byte array. There is a gas cost associated with accessing the byte at offset OFFSET of an execution environment's RAM:

$$C_{\text{mem}}(a) = G_{\text{mem}} \cdot a + \left\lfloor \frac{a^2}{512} \right\rfloor$$

where $G_{\text{mem}} = 3$ and the byte at byte offset OFFSET is contained in the a^{th} EVM word of RAM. In other words

$$a = 1 + \lfloor \mathsf{OFFSET}/32 \rfloor = \lceil (\mathsf{OFFSET} + 1)/32 \rceil$$

In other words, if we represent the byte array of RAM as below

OFFSET	0	1	$\cdots$	31	32	33	$\cdots$	63	64	65	$\cdots$	95	$\cdots$
EVM WORD	$a = 1$				$a = 2$				$a = 3$				

2 Constraints

2.1 Heartbeat

As already indicated, the three columns $\langle \mathsf{EXP\_STAMP} \rangle$, OOOB and CT define the heartbeat of the memory expansion module. Here are the constraints they satisfy:

1. OOOB is a binary column;
2. $\langle \mathsf{EXP\_STAMP} \rangle_0 = 0$;
3. $\langle \mathsf{EXP\_STAMP} \rangle$ is nondecreasing, i.e. $\forall i,\ \langle \mathsf{EXP\_STAMP} \rangle_{i+1} \in \{\langle \mathsf{EXP\_STAMP} \rangle_i, 1+\langle \mathsf{EXP\_STAMP} \rangle_i\}$;
4. $\textsc{if}$ $\langle \mathsf{EXP\_STAMP} \rangle_i = 0$ $\textsc{then}$ $\Big(\mathsf{OOOB}_i = 0 \ \textsc{and}\ \mathsf{CT}_{i+1} = \mathsf{CT}_i = 0\Big)$;
5. $\textsc{if}$ $\langle \mathsf{EXP\_STAMP} \rangle_i \neq 0$ $\textsc{then}$
   (a) $\textsc{if}$ $\mathsf{OOOB}_i = 0$ $\textsc{then}$
      i. $\textsc{if}$ $\mathsf{CT}_i \neq 2$ $\textsc{then}$
      $$\begin{cases} \mathsf{CT}_{i+1} = 1 + \mathsf{CT}_i \\ \langle \mathsf{EXP\_STAMP} \rangle_{i+1} = \langle \mathsf{EXP\_STAMP} \rangle_i \\ \mathsf{OOOB}_{i+1} = \mathsf{OOOB}_i \end{cases}$$
      ii. $\textsc{if}$ $\mathsf{CT}_i = 2$ $\textsc{then}$
      $$\begin{cases} \mathsf{CT}_{i+1} = 0 \\ \langle \mathsf{EXP\_STAMP} \rangle_{i+1} = 1 + \langle \mathsf{EXP\_STAMP} \rangle_i \end{cases}$$
   (b) $\textsc{if}$ $\mathsf{OOOB}_i = 1$ $\textsc{then}$
      i. $\textsc{if}$ $\mathsf{CT}_i \neq 16$ $\textsc{then}$
      $$\begin{cases} \mathsf{CT}_{i+1} = 1 + \mathsf{CT}_i \\ \langle \mathsf{EXP\_STAMP} \rangle_{i+1} = \langle \mathsf{EXP\_STAMP} \rangle_i \\ \mathsf{OOOB}_{i+1} = \mathsf{OOOB}_i \end{cases}$$
      ii. $\textsc{if}$ $\mathsf{CT}_i = 16$ $\textsc{then}$
      $$\begin{cases} \mathsf{CT}_{i+1} = 0 \\ \langle \mathsf{EXP\_STAMP} \rangle_{i+1} = 1 + \langle \mathsf{EXP\_STAMP} \rangle_i \end{cases}$$
6. $\textsc{if}$ $\langle \mathsf{EXP\_STAMP} \rangle_N \neq 0$ $\textsc{then}$
   (a) $\textsc{if}$ $\mathsf{OOOB}_N = 0$ $\textsc{then}$ $\mathsf{CT}_N = 2$;
   (b) $\textsc{if}$ $\mathsf{OOOB}_N = 1$ $\textsc{then}$ $\mathsf{CT}_N = 16$.

   In other words the module doesn't terminate mid instruction.

We say that a column X is **counter-constant** if it satisfies

$$\forall i,\ \mathsf{CT}_i \neq 0 \implies \mathsf{X}_i = \mathsf{X}_{i-1}.$$

Note that $\langle \mathsf{EXP\_STAMP} \rangle$ and OOOB are counter-constant by construction. We furthermore ask that *all* imported columns be counter-constant.

2.2 Offsets are in bounds

This section computes memory expansion in case both maximal offsets are in bounds. The first point is to establish this bound assertion. We then compare the two maximal offsets and store the greatest of the two in $\mathsf{MAX\_OFFSET}^{1,2}$.

All constraints in this section assume that in the current counter-cycle

$$\mathsf{EXP\_STAMP}_i \neq 0 \;\; \text{AND} \;\; \mathsf{OOOB}_i = 0.$$

2.2.1 Bound verification

1. $\mathsf{MAX\_OFFSET}^1$ and $\mathsf{MAX\_OFFSET}^2$ are 3 byte integers:

   (a) IF $\mathsf{CT}_i = 0$ THEN
   $$\begin{cases} \mathsf{ACC}^1_i = \mathsf{BYTE}^1_i \\ \mathsf{ACC}^2_i = \mathsf{BYTE}^2_i \end{cases}$$

   (b) IF $\mathsf{CT}_i \neq 0$ THEN
   $$\begin{cases} \mathsf{ACC}^1_i = 256 \cdot \mathsf{ACC}^1_{i-1} + \mathsf{BYTE}^1_i \\ \mathsf{ACC}^2_i = 256 \cdot \mathsf{ACC}^2_{i-1} + \mathsf{BYTE}^2_i \end{cases}$$

   (c) IF $\mathsf{CT}_i = 2$ THEN
   $$\begin{cases} \mathsf{ACC}^1_i = \mathsf{MAX\_OFFSET}^1_i \\ \mathsf{ACC}^2_i = \mathsf{MAX\_OFFSET}^2_i \end{cases}$$

2.2.2 Comparison of max offsets

1. COMP is a counter-constant binary column;

2. We ask that $\mathsf{COMP} = 1$ *iff* $\mathsf{MAX\_OFFSET}^1 \geq \mathsf{MAX\_OFFSET}^2$ (and thus $\mathsf{COMP} = 0$ *iff* $\mathsf{MAX\_OFFSET}^1 < \mathsf{MAX\_OFFSET}^2$). We ensure this like so:

   (a) IF $\mathsf{CT}_i = 0$ THEN $\Delta\_\mathsf{ACC}_i = \Delta\_\mathsf{BYTE}_i$

   (b) IF $\mathsf{CT}_i \neq 0$ THEN $\Delta\_\mathsf{ACC}_i = 256 \cdot \Delta\_\mathsf{ACC}_{i-1} + \Delta\_\mathsf{BYTE}_i$

   (c) IF $\mathsf{CT}_i = 2$ THEN

      i. IF $\mathsf{COMP}_i = 1$:
      $$\Delta\_\mathsf{ACC}_i = \mathsf{MAX\_OFFSET}^1_i - \mathsf{MAX\_OFFSET}^2_i$$

      ii. IF $\mathsf{COMP}_i = 0$:
      $$\Delta\_\mathsf{ACC}_i = \mathsf{MAX\_OFFSET}^2_i - \mathsf{MAX\_OFFSET}^1_i - 1$$

   In other words:
   $$\Delta\_\mathsf{ACC}_i = \left(\mathsf{MAX\_OFFSET}^1_i - \mathsf{MAX\_OFFSET}^2_i\right) \cdot (2 \cdot \mathsf{COMP}_i - 1) + (\mathsf{COMP}_i - 1)$$

3. we set $\mathsf{MAX\_OFFSET}^{1,2}_i$ to be the maximum of the two max offsets:
$$\mathsf{MAX\_OFFSET}^{1,2}_i = \mathsf{COMP}_i \cdot \mathsf{MAX\_OFFSET}^1_i + (1 - \mathsf{COMP}_i) \cdot \mathsf{MAX\_OFFSET}^2_i$$

2.2.3 Memory expansion recognition

1. $\mathsf{EXP\_FLAG}$ is a counter-constant binary column; we ask that $\mathsf{EXP\_FLAG} = 1$ *iff* $\mathsf{MAX\_OFFSET}^{1,2} > \mathsf{MEM\_SIZE}$; the following constraints ensure this:
   (a) IF $\mathsf{CT}_i = 0$ THEN $\mathsf{EXP\_ACC}_i = \mathsf{EXP\_BYTE}_i$
   (b) IF $\mathsf{CT}_i \neq 0$ THEN $\mathsf{EXP\_ACC}_i = 256 \cdot \mathsf{EXP\_ACC}_{i-1} + \mathsf{EXP\_BYTE}_i$
   (c) IF $\mathsf{CT}_i = 2$ THEN
      i. IF $\mathsf{EXP\_FLAG}_i = 1$:
      $$\mathsf{EXP\_ACC}_i = \mathsf{MAX\_OFFSET}_i^{1,2} - \mathsf{MEM\_SIZE}_i - 1$$
      ii. IF $\mathsf{EXP\_FLAG}_i = 0$:
      $$\mathsf{EXP\_ACC}_i = \mathsf{MEM\_SIZE}_i - \mathsf{MAX\_OFFSET}_i^{1,2}$$

   In other words when $\mathsf{CT}_i = 2$:
   $$\mathsf{EXP\_ACC}_i = \left(\mathsf{MAX\_OFFSET}_i^{1,2} - \mathsf{MEM\_SIZE}_i\right) \cdot (2 \cdot \mathsf{EXP\_FLAG}_i - 1) - \mathsf{EXP\_FLAG}_i.$$
2. IF $\mathsf{EXP\_FLAG}_i = 0$ THEN
   $$\begin{cases} \mathsf{MEM\_SIZE}_i^\nu = \mathsf{MEM\_SIZE}_i, \\ \mathsf{EXP\_COST}_i^\nu = \mathsf{EXP\_COST}_i, \end{cases}$$
3. IF $\mathsf{EXP\_FLAG}_i = 1$ THEN $\mathsf{MEM\_SIZE}_i^\nu = \mathsf{MAX\_OFFSET}_i^{1,2}$

We compute the updated expansion cost in the following section.

2.2.4 Memory expansion cost update

We compute the updated expansion cost. The following constraints apply *iff* $\mathsf{EXP\_FLAG} = 1$ in the current counter-cycle.

1. $\mathsf{QUOT\_1}$ is counter-constant;
2. $\mathsf{QUOT\_2}$ is counter-constant;
3. IF $\mathsf{EXP\_FLAG}_i = 1$:
   (a) IF $\mathsf{CT}_i = 0$ THEN $\mathsf{QUOT\_1\_ACC}_i = \mathsf{QUOT\_1\_BYTE}_i$
   (b) IF $\mathsf{CT}_i \neq 0$ THEN $\mathsf{QUOT\_1\_ACC}_i = 256 \cdot \mathsf{QUOT\_1\_ACC}_i + \mathsf{QUOT\_1\_BYTE}_i$
   (c) IF $\mathsf{CT}_i = 2$ THEN
   $$\begin{cases} \mathsf{QUOT\_1\_ACC}_i = \mathsf{QUOT\_1}_i \\ \mathsf{AUX\_1}_{i-1} = \mathsf{AUX\_1}_i + (256 - 32) & (1) \\ \mathsf{MEM\_SIZE}_i^\nu = 32 \cdot \mathsf{QUOT\_1}_i + \mathsf{AUX\_1}_i & (2) \end{cases}$$

   (1) verifies that the byte $\mathtt{r} := \mathsf{AUX\_1}_i$ is in the range $\{0, 1, \ldots, 31\}$; (2) verifies that the 3 byte integer $\mathsf{QUOT\_1}_i$ and $\mathtt{r}$ are the quotient and residue respectively of the euclidean division of $\mathsf{MEM\_SIZE}_i^\nu$ by 32; together they verify Eq. (♣).
   (d) IF $\mathsf{CT}_i = 0$ THEN $\mathsf{QUOT\_2\_ACC}_i = \mathsf{QUOT\_2\_BYTE}_i$
   (e) IF $\mathsf{CT}_i \neq 0$ THEN $\mathsf{QUOT\_2\_ACC}_i = 256 \cdot \mathsf{QUOT\_2\_ACC}_i + \mathsf{QUOT\_2\_BYTE}_i$

(f) IF $\mathsf{CT}_i = 2$ THEN

$$\begin{cases} \mathsf{AUX\_2}_{i-2}^2 = \mathsf{AUX\_2}_{i-2} & (1) \\ \mathsf{QUOT\_2\_ACC}_i + 256^3 \cdot \mathsf{AUX\_2}_{i-1} = \mathsf{QUOT\_2}_i & (2) \\ (1 + \mathsf{QUOT\_1}_i)^2 = 512 \cdot \mathsf{QUOT\_2}_i + 256 \cdot \mathsf{AUX\_2}_{i-2} + \mathsf{AUX\_2}_i & (3) \end{cases}$$

(1) verifies that the byte $\epsilon := \mathsf{AUX\_2}_{i-2}$ is actually a bit; (2) verifies that the byte $\mathtt{b}_3 := \mathsf{AUX\_2}_{i-1}$ is the most significant byte of the 4 byte integer $\mathsf{QUOT\_2}_i$[2]; (3) verifies that $256 \cdot \epsilon + \mathtt{b}$ is the remainder of the euclidean division of $(1 + \mathsf{QUOT\_1}_i)^2$ by 512, where $\mathtt{b} := \mathsf{AUX\_2}_i$; together they verify Eq. (♠).

(g) $\mathsf{EXP\_COST}_i^\nu = 200 \cdot (1 + \mathsf{QUOT\_1}_i) + \mathsf{QUOT\_2}_i$

(h) verify the gas expansion cost:

$$\langle \mathsf{EXP\_GAS}_i \rangle = \mathsf{EXP\_COST}_i^\nu - \mathsf{EXP\_COST}_i.$$

2.3 Offsets are grossly out of bounds

This section is about raising an out of gas error early when one of the max offsets is grotesquely out of bounds. All constraints in this section assume that in the current counter-cycle

$$\boxed{\mathsf{EXP\_STAMP}_i \neq 0 \ \text{ AND } \mathsf{OOOB}_i = 1.}$$

2.3.1 Bound verification

1. IF $\mathsf{EXP\_STAMP}_i \neq 0$ AND $\mathsf{OOOB}_i = 1$ THEN one of $\mathsf{MAX\_OFFSET}^1$, $\mathsf{MAX\_OFFSET}^2$ isn't a 3 byte integer:

   (a) IF $\mathsf{CT}_i = 0$ THEN $\mathsf{ACC}_i^1 = \mathsf{BYTE}_i^1$;

   (b) IF $\mathsf{CT}_i \neq 0$ THEN $\mathsf{ACC}_i^1 = 256 \cdot \mathsf{ACC}_{i-1}^1 + \mathsf{BYTE}_i^1$;

   (c) IF $\mathsf{CT}_i = 16$ THEN

   $$\Big((\mathsf{MAX\_OFFSET}_i^1 - 256^3) - \mathsf{ACC}_i^1 \Big) \cdot \Big((\mathsf{MAX\_OFFSET}_i^2 - 256^3) - \mathsf{ACC}_i^1 \Big) = 0$$

   in other words one of $(\mathsf{MAX\_OFFSET}_i^1 - 256^3)$ and $(\mathsf{MAX\_OFFSET}_i^2 - 256^3)$ is a 17 byte integer.

2.4 Consistency constraints

We impose some consistency constraints on the memory expansion module. For this we consider a row reordering $\mathsf{X} \rightsquigarrow \widetilde{\mathsf{X}}$ such that the following columns are listed in lexicographic order:

$$\left(\widetilde{\mathsf{CN}}, \widetilde{\mathsf{EXP\_STAMP}} \right)$$

We say *a* row reordering since the module may start with an arbitrary number of empty columns. Besides the inconsequential order on those, the ordering is unique.

1. IF $\widetilde{\mathsf{CN}}_i \neq 0$:

   (a) IF $\widetilde{\mathsf{CN}}_{i+1} = \widetilde{\mathsf{CN}}_i$ THEN

   $$\begin{cases} \mathsf{MEM\_SIZE}_{i+1} = \mathsf{MEM\_SIZE}_i^\nu \\ \mathsf{EXP\_COST}_{i+1} = \mathsf{EXP\_COST}_i^\nu \end{cases}$$

[2] the other bytes being, in increasing order of significance: $\mathsf{QUOT\_2\_BYTE}_{i-2}$, $\mathsf{QUOT\_2\_BYTE}_{i-1}$ and $\mathsf{QUOT\_2\_BYTE}_i$

(b) IF $\widetilde{\mathsf{CN}}_{i+1} \neq \widetilde{\mathsf{CN}}_i$ THEN

$$\begin{cases} \mathsf{MEM\_SIZE}_{i+1} = 0 \\ \mathsf{EXP\_COST}_{i+1} = 0 \end{cases}$$